dSPACE MAGAZINE

2/2014

Tetra Pak –
Virtual Packaging Plant

PSA – Driving AUTOSAR

BMW Motorrad –
Electronic Cornering Grip


dSPACE
26

Since the global economic crisis, particularly in the automotive industry in 2008-2009, development activity has picked up speed again, and unexpectedly fast. This, of course, increased business for companies such as dSPACE.

It was clear that the industry had to make up for lost time. We also expected that our customers would not have enough development resources to actually bring their many new projects and on-hold projects to reality. This, in turn, gave engineering service providers the opportunity to step in, but they also reached their maximum capacity. Tool providers were therefore well-advised to expect slow or zero growth in 2013, which most of them indeed experienced. But this year, the winds have changed. Investments have gone up. The curve is pointing upward. Nevertheless, many large automobile manufacturers are talking about cutting costs again. Let’s hope they know where to cut them. Not investing in improved development efficiency or complexity management would certainly not be a good idea.

dSPACE has to meet many requirements from the automotive industry, but regular editorial readers surely remember that we also care about non-automotive applications. The last dSPACE Magazine issue presented how hardware-in-the-loop (HIL) simulation is used for food service equipment, such as beverage dispensers. This issue brings you an example from the beverage packaging industry. It has long been known that developers of process control software can also benefit from such modern development methods. But practical applications were rare, because the step of going from “We have to install the plant anyway, so we’ll test it directly at the client.” to introducing a HIL simulator requires great effort. This should not be underestimated. But it is a step worth taking. Because once the plant is installed, it is under constant, close scrutiny and every day it does not function properly is a very expensive day. We are eager to see if breweries and wineries follow the example set by providers of beverage and milk filling lines and rely on dSPACE in the future. We’ll come and have a test.

Dr. Herbert Hanselmann
President

Embedded Success dSPACE

IMPRINT

dSPACE MAGAZINE is published periodically by:

dSPACE GmbH · Rathenaustraße 26
33102 Paderborn · Germany
Tel.: +49 5251 1638-0
Fax: +49 5251 16198-0
dspace-magazine@dspace.com
www.dspace.com

Responsible in terms of press law:
Bernd Schäfers-Maiwald
Project Manager: André Klein

Authors:
Thorsten Bödeker, Ralf Lieberwirth, Sonja Lillwitz, Thomas Pöhlmann, Julia Reinbach, Dr. Gerhard Reiß

Co-Workers on this issue:
Michael Beine, Dr. Ulrich Eisemann, Gregor Hordys, Doreen Krob, Andre Rolfsmeier, Thomas Sander, Joachim Stroop, Dr. Ingo Stürmer

Editors and Translators:
Robert Bevington, Stefanie Bock, Anna-Lena Huthmacher, Dr. Michelle Kloppenburg, Christine Smith

Design and Layout:
Jens Rackow, Sabine Stephan

Print:
Media-Print Group GmbH, Paderborn

Contents

Customers

Products

Business

PEFC certified
The paper for this magazine is from sustainably managed forests and controlled sources.
www.pefc.org
PEFC/04-31-0810

ClimatePartner
climate neutral
Print | ID: 53446-1408-1004

To optimize cooperation with the electronic control unit suppliers, PSA decided to make the development process for drivetrain ECUs AUTOSAR-compliant. To do this efficiently, PSA set up a tool chain that seamlessly supports elementary process steps from creating the architecture to actual implementation. Several AUTOSAR-compliant production projects have since been successfully completed.

AUTOSAR Implemented

AUTOSAR-compliant production software development with TargetLink and SystemDesk

The software in PSA engine ECUs supports functions for combustion engines, hybrid controllers and transmission applications.

Established Model-Based Tool Chain Based on TargetLink/Simulink

TargetLink®, dSPACE's production code generator, has been used at PSA for production software development in the drivetrain area since 2007 – for combustion engines, hybrid controllers, and transmission applications.

The reasons for using automatic production code generation were classic factors such as time to market, the need to handle the increasing complexity of automotive software, and the fact that other departments deliver models as executable specifications within PSA, where an implementation team turns them into production software. In-house benchmark tests with the first version of TargetLink to be used, 2.2.1, confirmed the expected benefits: an accelerated development process and higher software quality. The advent of the AUTOSAR standard gave PSA the opportunity to simplify cooperation with suppliers by putting it on a new, AUTOSAR-compliant basis. This was the main motivation for extending the tool chain to support AUTOSAR-compliant development and for using standardized AUTOSAR-XML (ARXML) exchange formats instead of proprietary formats. Other tools were used for this in addition to the TargetLink AUTOSAR Module, particularly the system architecture tool dSPACE SystemDesk®.

From Requirement to Implementation

Software requirements are the starting point and the basis for development. They can be included in the workflow in one of two defined ways, depending on the project.

1. *Simulink/TargetLink Models*
   Requirements are described in the form of models that PSA develops internally and then passes to another team for software implementation. In this case the model serves as an executable specification of the algorithmic behavior, which is then converted into software by adding software properties with TargetLink.
2. *Textual Requirements*
   Requirements can also be specified in classic text form, for example, in IBM® Rational® DOORS®. In this case the implementation specialists perform the actual function/algorithm development with TargetLink and also the software implementation from the models that they create.

AUTOSAR-Compliant Development Process

The architecture of the ECUs' application software is described with SystemDesk in compliance with AUTOSAR. Currently at PSA, the software architecture for engine ECUs consists of 70 AUTOSAR application components with approx. 3000 interfaces and 300 runnables. All the AUTOSAR specifications, including the parts needed to generate a run-time environment (RTE), are completely available in SystemDesk. Either this data is created manually in SystemDesk, or existing interfaces or runnable specifications are imported from the Visu-IT! Automotive Data Dictionary (ADD) or from existing models.
To develop the functions of a software component, first its AUTOSAR

> "We have been using TargetLink since 2007 to develop software for engine management systems with a model-based design approach. This tool provides efficiency in our development process and offers features that we need to produce AUTOSAR-compliant code."
>
> *Nabile Khoury, PSA*

properties are exported from SystemDesk (AUTOSAR ARXML files) and transferred to the TargetLink Data Dictionary. The TargetLink user generates an AUTOSAR frame model from them and inserts the algorithmic functionality under development into the frame model (this is the top-down approach). Currently, AUTOSAR standard version 3.1.2 is being used, and all PSA components are implemented with TargetLink.

Model Design and Automatic Code Generation

TargetLink models for implementing individual AUTOSAR software components are designed according to the following rules:

- The models must be compatible with the software architecture in SystemDesk. Compliance with this rule is practically automatic with the AUTOSAR-compliant workflow and AUTOSAR frame model generation.
- The models must implement the software requirements. For functional requirements, this rule is generally satisfied if the specifications are supplied as models. Such models just have to be converted from Simulink® to TargetLink if necessary.
  If the requirements are in text form, the model is validated against them by offline simulations and other methods.

The individual TargetLink models for implementing the functionality of individual components can vary greatly in size, with the largest SWC models containing up to 6000 computation blocks. For the actual control design, a proprietary PSA library is used in addition to TargetLink blocks. The library contains functionalities such as counters and filters, and TargetLink's scaling invariance features are also partly used.
To ensure high model quality, modeling guidelines from the following bodies are applied: MathWorks®, dSPACE, and PSA.
Either floating-point or fixed-point code is generated depending on the project, and also on the complexity and required precision.

The software for PSA engine ECUs is developed with TargetLink and SystemDesk.

Test Activities for Components and Integrations

Because PSA bears complete responsibility for its own software, the individual software components are tested in TargetLink software-in-the-loop (SIL) and processor-in-the-loop (PIL) simulations, supported by BTC EmbeddedTester®:

> "SystemDesk helps us to efficiently design the software architecture of our engine management systems with AUTOSAR constraints. Its integrated environment also enables us to verify that our applications can be integrated in AUTOSAR platforms."
>
> *Zhao Zuo, PSA*

- *Automatic test vector generation and back-to-back testing with BTC EmbeddedTester*
  MIL/SIL/PIL simulation results are compared automatically with the help of automatic test vector generation by the Embedded Tester and so-called back-to-back tests. This is particularly helpful in the case of supplied controller models, in order to validate automatically that the controller model and the software implementation with TargetLink have sufficiently similar behaviors. Modified condition/decision coverage (MC/DC) measurements are also performed to verify that a satisfactory code coverage is achieved.
- *PIL tests for profiling the required ECU resources*
  With the help of TargetLink PIL simulation, a resource profile is created in order to estimate the memory (RAM/ROM/stack) that will be consumed at run time.
- *Functional tests with other data*
  Other test cases are developed manually on the basis of functional requirements or taken from recorded vehicle data. These tests are executed with Embedded Tester and an in-house test tool.

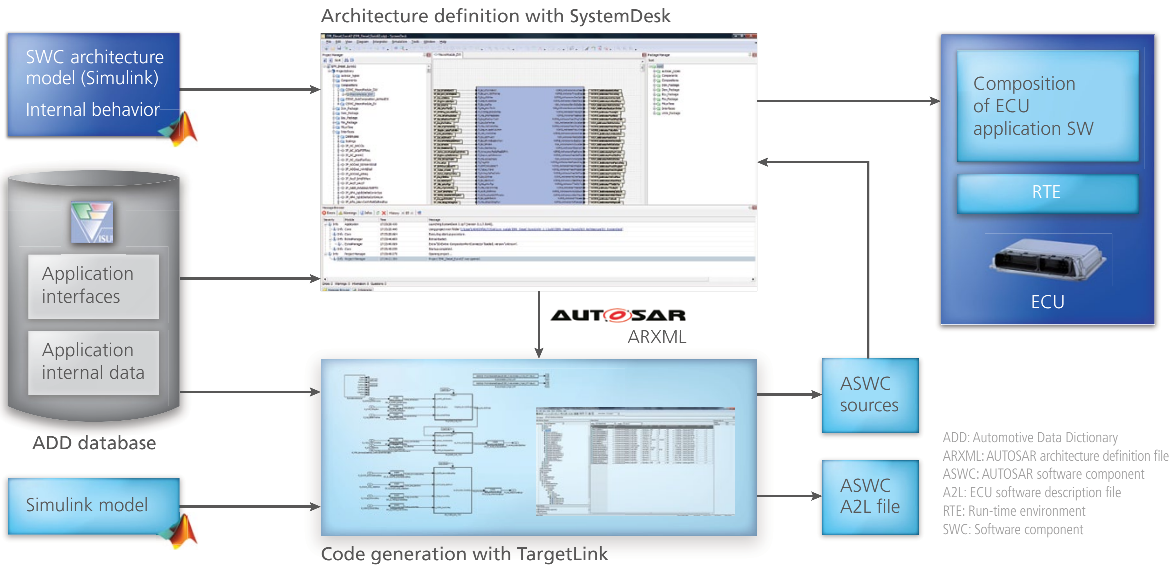
PSA not only tests components but also performs software integrations for test purposes. An RTE for the ECU's entire software architecture is generated from within SystemDesk and temporarily integrated with the TargetLink components (this is called pre-integration). This step ensures that no problems occur when the actual integration is performed later by the supplier.

The Roles of OEM and Suppliers

The division of tasks and exchange of artifacts between PSA and suppliers looks like this:

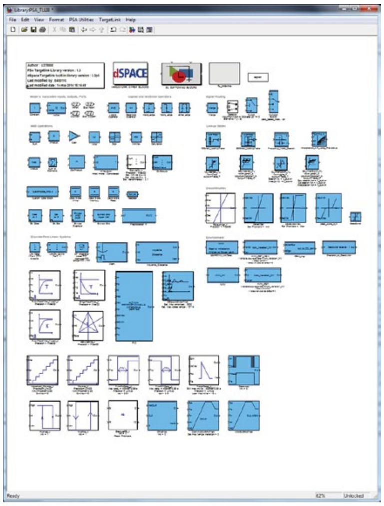
- PSA is responsible for the ECU's software architecture and passes

AUTOSAR workflow with SystemDesk and TargetLink.



it to the supplier as AUTOSAR ARXML specifications.

- PSA provides the major part of the application software for the ECUs and also thoroughly tests the individual software components as described above.
- Some of the software components are outsourced to external developers ("software as a product") and made directly available to PSA.
- For IP protection, the generated source code is intentionally made unreadable by a code obfuscator. It is therefore not necessary to exchange object code, and the ECU suppliers are free to choose their own compilers and compiler options.
- The ECU supplier also contributes small parts of the application software and uses RTE generation to integrate them and the components supplied by PSA on the ECU together with the basic software, after which the ECU is delivered to PSA.



PSA block library for recurring functionalities such as filters and counters.

Experience with Using the dSPACE AUTOSAR Tool Chain

Since starting to use the TargetLink/SystemDesk combination, PSA has successfully developed software for 10 engine ECUs, 1 transmission ECU, and 2 hybrid ECUs, and brought it up to production level. The proportion of the application software produced by PSA varies between 60% and 95%. The entire software is developed with TargetLink and SystemDesk, in other words, model-based design and autocoding are the established methods. For example, for an engine ECU, PSA produces approx. 1 megabyte of code, created by approx. 50 developers. A special focus of current projects is to implement the requirements resulting from the Euro 6.2 standard. This is being done with TargetLink 3.2 and SystemDesk 3.1, based on AUTOSAR Version 3.1.2.

Conclusion and Outlook

The AUTOSAR-compliant tool chain with SystemDesk and TargetLink has proved successful in several production projects. SystemDesk supports the convenient, efficient design of extensive software architectures. Two of TargetLink's indispensable strengths are its strong focus on software implementation tasks and its good integration with MATLAB®/Simulink® and EmbeddedTester. With the tool's high usability, the efficient and well-documented automation APIs, and the PSA-specific extensions, even new users can very quickly start working productively. In the future, PSA plans to migrate to AUTOSAR 4 with TargetLink 3.5 and SystemDesk 4.1. ■

Nabile Khoury, Zhao Zuo,
PSA Peugeot Citroën

Summary

French carmaker PSA Peugeot Citroën develops the software for drivetrain ECUs of all their model series in compliance with AUTOSAR. The company creates most of the application software itself, using a tool chain that consists of the architecture software SystemDesk, the model-based development environment MATLAB/Simulink, and the production code generator TargetLink. The functions that are developed are first implemented on an ECU and thoroughly tested by PSA. Then the ECU supplier integrates the tested PSA software components, parts of the application software that they have developed themselves, and the basic software on the production ECU. The efficiency of the development process is ensured by standardized exchange formats, clearly structured components, and the seamless tool chain.

Nabile Khoury
Nabile Khoury is Model-Based-Design Specialist at PSA Peugeot Citroën in La Garenne-Colombes, France.

Zhao Zuo
Zhao Zuo is AUTOSAR Software Architect at PSA Peugeot Citroën in La Garenne-Colombes, France.

Developing innovative features for vehicle dynamics at the extremes requires the right tools that do not manipulate vehicle behavior. BMW Motorrad has found the perfect solution.

Reality Is Key

BMW Motorrad relies on model-based development and rapid control prototyping (RCP) at the vehicle itself to develop new functions. The development team uses MATLAB®/Simulink® to develop new controllers, such as anti-lock braking systems (ABS), automatic stability control (ASC), dynamic traction control (DTC), or dynamic damping control (DDC). Basic functional aspects can be analyzed in offline simulation.

But there are complex influencing factors, such as the driver's behavior, the tires, and the fact that vehicle dynamics are often taken to the extremes. A controller's quality can therefore only be evaluated on the real vehicle. The development department for brake and control systems at BMW Motorrad, Function Development, uses both dedicated RCP systems and production ECUs with certain development software versions for in-

Maximum Tilt

Developing and testing vehicle dynamics controllers on motorcycles

vehicle testing. Both of these approaches have their advantages and drawbacks.

Finding the Right Balance

High processing power, a flexible I/O and comprehensive, intuitive configuration options make it possible to develop RCP systems fast and easily, minimizing the iteration cycles. But there are other aspects hampering the use of RCP systems. For example, if an RCP system is installed on a motorcycle, it changes the vehicle's center of gravity, affecting its driving behavior during test drives and causing a crucial problem for vehicle dynamics development. Having to install and uninstall an RCP system also makes it more difficult to change test vehicles. Last but not least, RCP systems are used under extremely adverse conditions. One major challenge even for production ECUs is the vibrations an engine causes when it is mounted directly on the chassis and reaches a rotational speed up to 14,000 rpm. This is why function development for testing and prototyping often uses the motorcycle's own production ECU.

Production Before Predevelopment

Since design engineers cannot directly access the development environment and source code of the ECU software, they cooperate with the software department responsible for production development. The software department uses the C code of new Simulink

Top athlete BMW S 1000 RR uses a dynamic traction control system (DTC) to render maximum power under any road condition.
The DTC monitors the wheel speed of the rear wheel and the information given by the sensor box (such as tilt information).
It also limits the engine torque depending on rear tire slip to prevent the motorcycle from swerving or sliding.
This improves driver safety during sporty maneuvers even when road conditions are difficult.

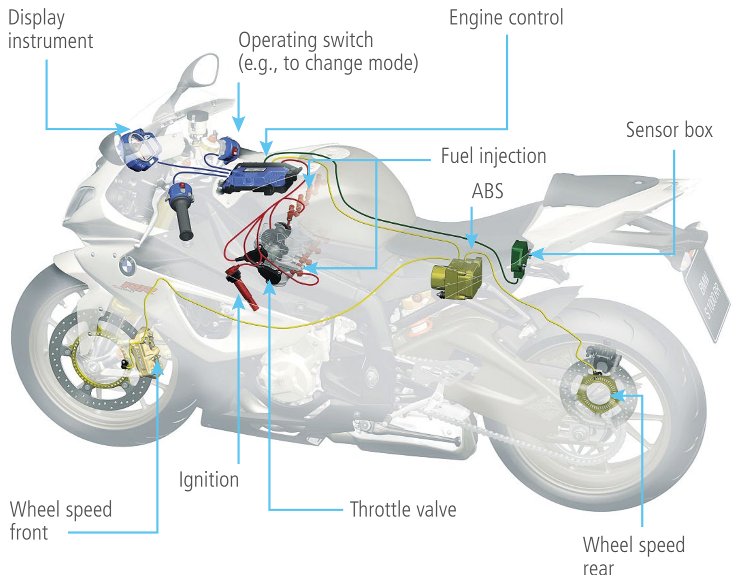


Figure 1: System design of the dynamic traction control system.

> “On-target prototyping by dSPACE is an entirely new development method that makes us independent, saving both effort and time. We can continue to focus on innovation.”
>
> *Martin Heidrich, BMW Motorrad*

functions to develop a software version tailored to the needs of function development. The great advantage is that these software versions can easily be flashed onto different ECU types, allowing for a quick change of test vehicles. Relatively long lead times and iteration cycles are a drawback of this approach. And production usually has priority over predevelopment. Furthermore, the memory available for development purposes limits the number of controller variants that can be integrated simultaneously. Comparing several controller variants therefore becomes more difficult. The ideal solution would be to develop directly on an ECU designed for use on motorcycles, while benefiting from established RCP tool chains.

The New Approach: On-Target Prototyping

The ECU Interface Manager and RTI Bypass Blockset from dSPACE form the tool chain that makes rapid control prototyping on production ECUs possible. BMW Motorrad decided to test this tool chain in a new project to further develop the dynamic traction control system (DTC) of the BMW S 1000 RR (figure 1).

On-Target Prototyping in Practice

Largely independently from production development, design engineers took new functions developed in MATLAB/Simulink and integrated them directly into the HEX code of a production software used in a project (figure 2).
It was not necessary to access the source code or the build environment of the development software. The only configuration information needed from the software department was information that is available anyway during production software development. The modified software versions were then easily flashed onto the ECUs of various test vehicles to validate their functions during test drives. Developers used the test results to

improve the functions and test these on the vehicle using new software versions. This fast, iterative process allowed developers to test a greater number of controller variants, change test vehicles more quickly, and bring the new functions to greater maturity, while fully maintaining the ease of use of RCP systems. This approach's flexibility made it possible to quickly implement even short-term changes ordered by the management.

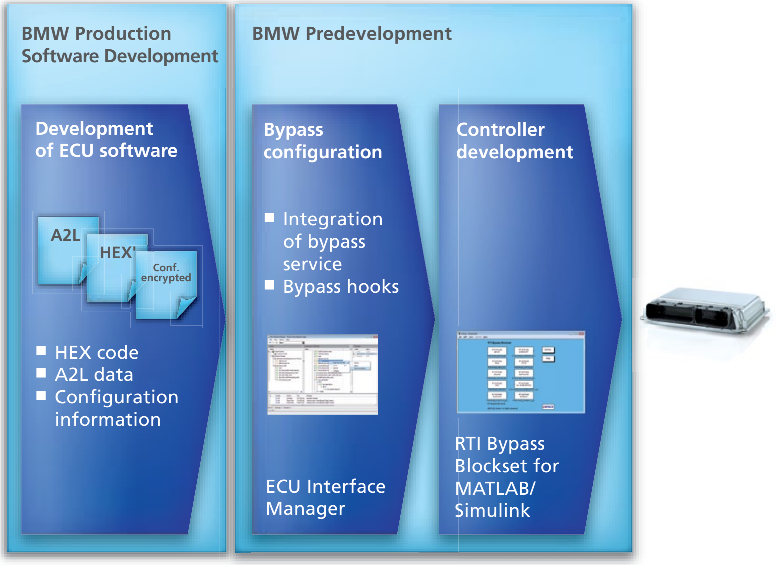


Figure 2: The workflow with the ECU Interface Manager. New functions can be integrated directly into the production ECU's software.

White-Box Testing on Target

But on-target prototyping benefits not only development itself. Soon, design engineers realized that this new tool chain can also be used for the final approval test for new functions. Design engineers are responsible not only for developing new functions but also for approving and releasing the production code implemented by the software department. Since input and output values of the new functions can be accessed on the ECU itself, design engineers were able to carry out white-box testing of the production implementation under real-time conditions. External stimulation of I/O and the use of real sensors and tedious restbus simulation became obsolete and a large part of approval tests were moved from the road to the lab. The result: reduced verification effort and a much shorter verification phase. ■

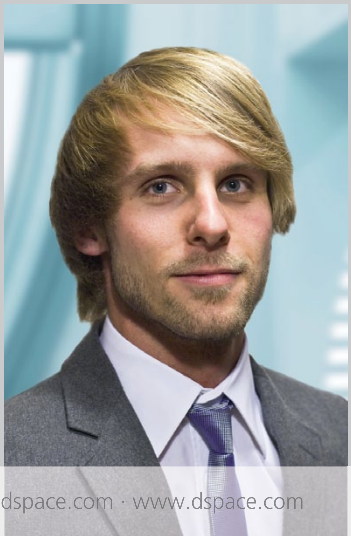
Martin Heidrich,
Josef Rank,
BMW Motorrad

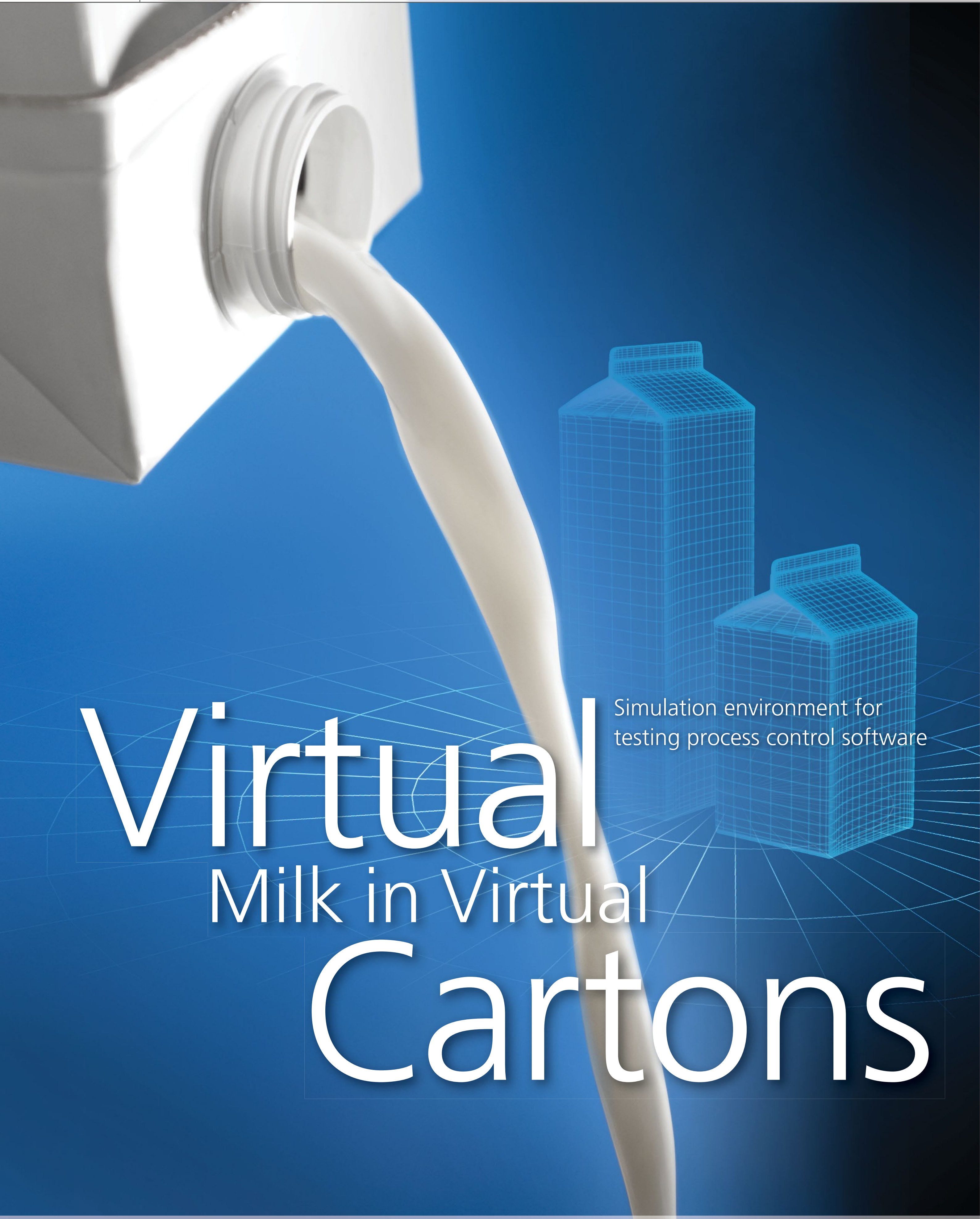
Conclusion

BMW Motorrad extended their development process with a new development method: on-target prototyping. This method, based on the dSPACE ECU Interface Manager and RTI Bypass Blockset, was evaluated in the development project and passed the test with flying colors. It is more flexible, shortens development cycles and helps reach a higher product maturity. BMW developers were able to focus on the core of BMW Motorrad – innovation.

Martin Heidrich
Martin Heidrich is design engineer and software developer for vehicle control systems at BMW Motorrad in Munich, Germany.

Josef Rank
Josef Rank is design and application engineer ASC/DTC at BMW Motorrad in Munich, Germany.





Virtual Milk in Virtual Cartons

Simulation environment for testing process control software

Intelligent process control software has an important job to do to continuously optimize downtimes, but developing and testing it is time-consuming and expensive. This issue is successfully addressed by the Tetra Pak Simulation Environment (TSE), an efficiency-boosting software and hardware system for the virtual commissioning of beverage filling machines.

Challenges for the Food Service Industry

In times of shifting markets, the food service industry is facing new challenges. Increasing cost pressure and intense competition are forcing companies to optimize their processes and shorten the time to market. This was one of the factors that prompted Tetra Pak to develop the Tetra Pak Simulation Environment (TSE) for the virtual testing of beverage filling machines. As a leading provider of processing and packaging systems for beverage products, Tetra Pak gained an international reputation with its milk cartons.

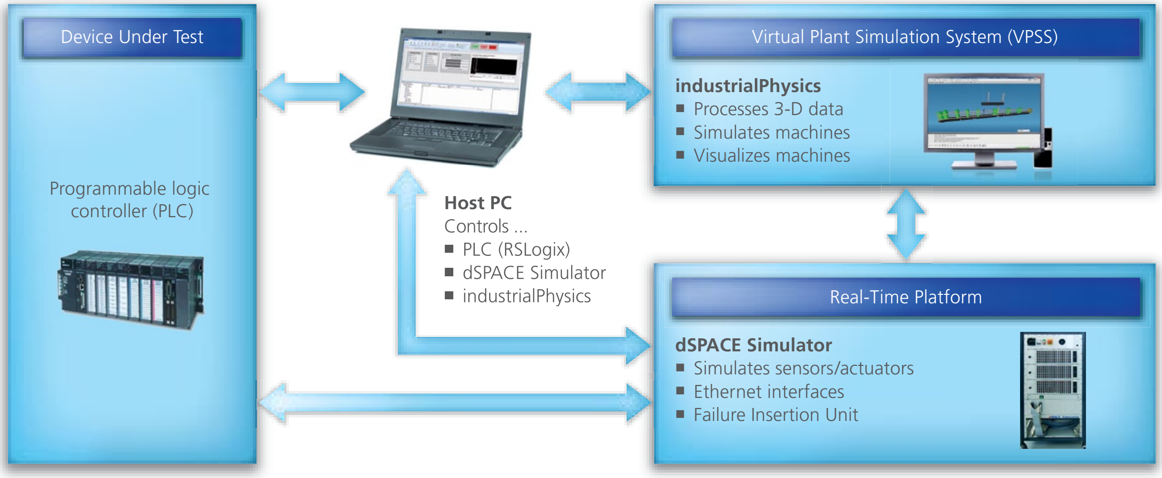
Broad Potential for Simulation

Tetra Pak has two primary objectives in developing the TSE. One is to support the development of process control software for filling machines. The TSE does this by simulating and optimizing new concepts before they even reach prototype status, and by enabling developers to test and optimize their own code.
The other is to run hardware-in-the-loop (HIL) simulations of the programmable logic controller (PLC), including the process control software, on the TSE. The trend toward HIL testing has recently gained momentum, especially in the German and Italian packaging industries, because virtual machine commissioning enables them to automate tests and run regression tests. The latter ensure that the software will function reliably under real-world conditions.

How the Simulation Environment Helps

The environment gives users decisive advantages. Firstly, by simulating a PLC's work environment, it greatly reduces the expense and time needed to construct real prototype machines. There is also no risk of damage to any hardware, which can occur if real

A Tetra Pak beverage filling machine: By the time the first prototype is ready, its process control has already passed a whole series of virtual tests.



Structure of the simulation environment: Simulation tools from dSPACE and the industrialPhysics software are the heart of the Tetra Pak Simulation Environment (TSE).

machines are used to test a function such as an emergency stop. And finally, simulation reduces the amount spent on liquid products and packaging materials.

With individually selectable add-ons, there is a wide range of different machine variants, and they all need thorough testing before mandatory or optional software updates can be supplied to the customer. Here too, HIL simulations help identify problems at an early stage. Furthermore, HIL-based software tests are just as important when customers want to integrate new components into existing machines.

Tetra Pak Seeks All-Round Talent

In the hunt for the optimum project partner, Tetra Pak evaluated the portfolio of services provided by various suppliers of HIL processes and automation solutions, and also those of machine manufacturers. It was very quickly clear that the ambitious vision could not be achieved with out-of-the-box solutions, rather it would have to be built step by step. Not least because of this, Tetra Pak placed particular importance on receiving sound engineering support over and beyond the usual standard products.

Solution Competence from dSPACE

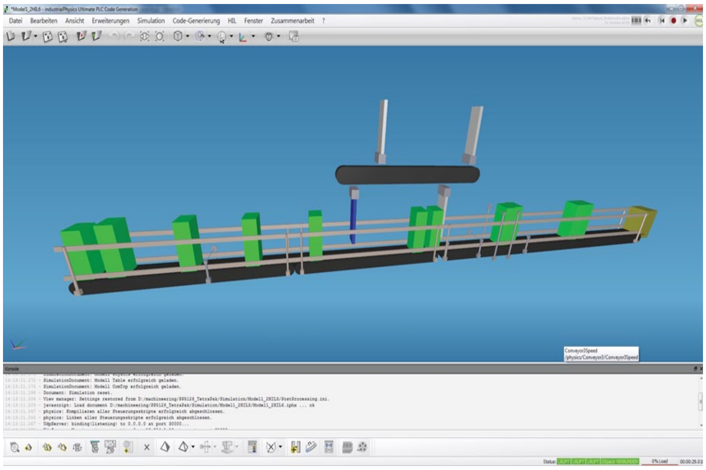
Tetra Pak chose dSPACE because of their well-recognized know-how on HIL and modeling. The company's versatile product portfolio covers almost the entire HIL tool chain that the project requires – with ControlDesk® Next Generation, Real-Time Interface for Multiprocessor Systems, the AutomationDesk test automation software, and so on. Powerful and scalable hardware for multiprocessor structures is also available.

Moreover, dSPACE has comprehensive experience in multiple sectors, acquired over many years in the automotive, aerospace, robotics and e-drive industries. dSPACE's location not far from Tetra Pak was also important for the project, as it makes close cooperation in the development process much easier.

Structure of the Simulation Environment

To produce the realistic simulation scenarios that Tetra Pak is aiming for, the TSE uses a dSPACE HIL simulator with two processor boards, analog

> "dSPACE's comprehensive know-how and product portfolio are the ideal foundation for building a simulation environment for Tetra Pak beverage filling machines."
>
> *Mauro Gargiulo, Tetra Pak*



```
14:18:11.172 - SimulationDocument: Modell Table erfolgreich geladen.
14:18:11.176 - SimulationDocument: Modell ComTrp erfolgreich geladen.
14:18:11.199 - Document: Simulation reset.
14:18:11.206 - View manager: Settings restored from D:/machineering/PP5128_TetraPak/Simulation/Modell_2HIL8/PostProcessing.ini.
14:18:11.209 - javascript: Load document D:/machineering/PP5128_TetraPak/Simulation/Modell_2HIL8/Modell_2HIL6.iphz ... ok
14:18:11.247 - physics: Kompilieren aller Steuerungsskripte erfolgreich abgeschlossen.
14:18:11.255 - physics: Linken aller Steuerungsskripte erfolgreich abgeschlossen.
14:18:11.247 - UdpServer: binding(listening) to 0.0.0.0 at port 30000...
```

Physical computations made by the industrialPhysics simulation tool produce a virtual image of the machine.

Summary

The Tetra Pak Simulation Environment effectively supports the development of control software for beverage filling machines long before the first prototype is constructed. The dSPACE HIL simulator is the heart of the system. It enables developers to test a PLC reliably under real-world conditions and helps to permanently boost machine efficiency.

and digital I/O boards, and a Failure Insertion Unit. Other components in the simulation environment are the host PC, the 3-D virtual plant simulation system (VPSS), and what is called the control environment, which contains the programmable process control to be validated and also the human-machine interface (HMI).

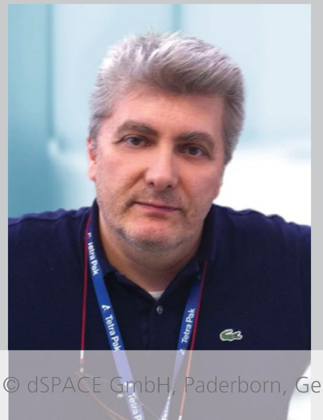
Visualization and Simulation

For testing purposes, dSPACE simulation tools and industrialPhysics (a tool from machineering GmbH & Co. KG) stand in for the real machine. Tasks such as 3-D computation and visualization of the virtual machine, package flow, and conveyor belts are performed by industrialPhysics, while the dSPACE tools are responsible for simulating the thermodynamics, fluid mechanics, and so on. Where strict real time and small simulation step sizes are required (for example, to simulate filling processes or perform motion control tasks), the dSPACE tools are indispensable. ■

Roberto Borsari,
Mauro Gargiulo, Tetra Pak
Georg Wünsch,
machineering GmbH & Co. KG

Mauro Gargiulo
Mauro Gargiulo heads the Systems Engineering department for Packaging Material at Tetra Pak Packaging Solutions AB in Lund, Sweden.

Roberto Borsari
Roberto Borsari is responsible for Forming & Virtual Verification at Tetra Pak Packaging Solutions in Modena, Italy.



Georg Wünsch
Georg Wünsch is in charge of development at machineering GmbH & Co. KG in Munich, Germany.

There are conventional aircraft and unmanned aerial vehicles, and a third category in between, called optionally piloted vehicles (OPVs). These are planes that can fly with or without a pilot as the situation requires. The Korea Aerospace Research Institute (KARI) is developing an OPV and testing its flight control system by running simulated flights in the laboratory with a test bench based on a dSPACE hardware-in-the-loop simulator.

Onboard view during a test flight. A complete preflight simulation of the mission in the laboratory is performed by a dSPACE simulator.

With or Without a Pilot

Developing an optionally piloted vehicle

Why Use OPVs?

The advantage is that OPVs can manage without a human crew on easy-to-navigate missions that do not require any on-the-spot decisions to be made by a pilot. Tedious, long observation missions are typical examples. Obviously, without a pilot on board, an OPV needs particularly mature (multi-redundant) flight control systems in order to fly autonomously and safely. KARI is developing algorithms for such flight control systems.

Realistic Flight Simulation

The flight control systems are being developed with the aid of test bench tests (i.e., virtual flights in the laboratory). Suitable MATLAB®/Simulink® models had to be developed for this. To reduce the differences between virtual and real flight, the flight dynamics model was validated with flight test data (figure 1).

Also, to aid pilot training, not just the flight itself, but also taking off, landing, and taxiing on the airfield was simulated. The virtual flights also have to cover different weather conditions (such as squalls). Using the validated flight dynamics model, all the automatic flight control laws were validated and tuned during the HIL phase. As a result, there was not any tuning necessary during unmanned flight tests. The performance matched that of virtual flights very well.

> “With the test station based on the dSPACE Simulator, we can test all the functions of the flight control system without the aircraft having to leave the ground.”
>
> *Dr. Hyoung-Sik Choi, KARI*

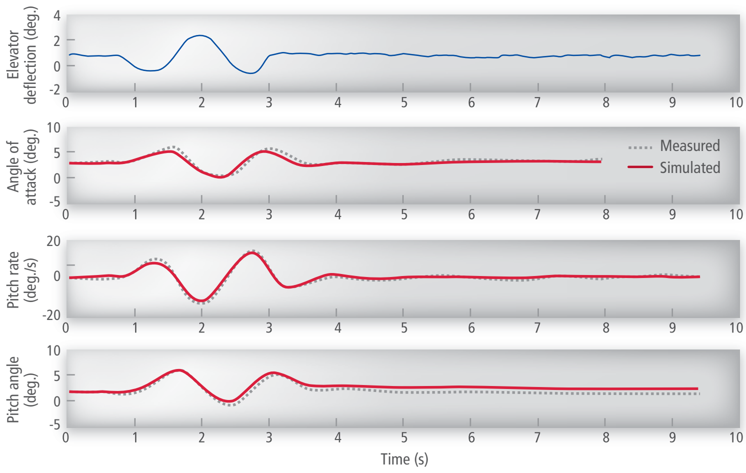


Figure 1: Flight dynamics model validation using flight test data (blue: elevator input; red: longitudinal response)

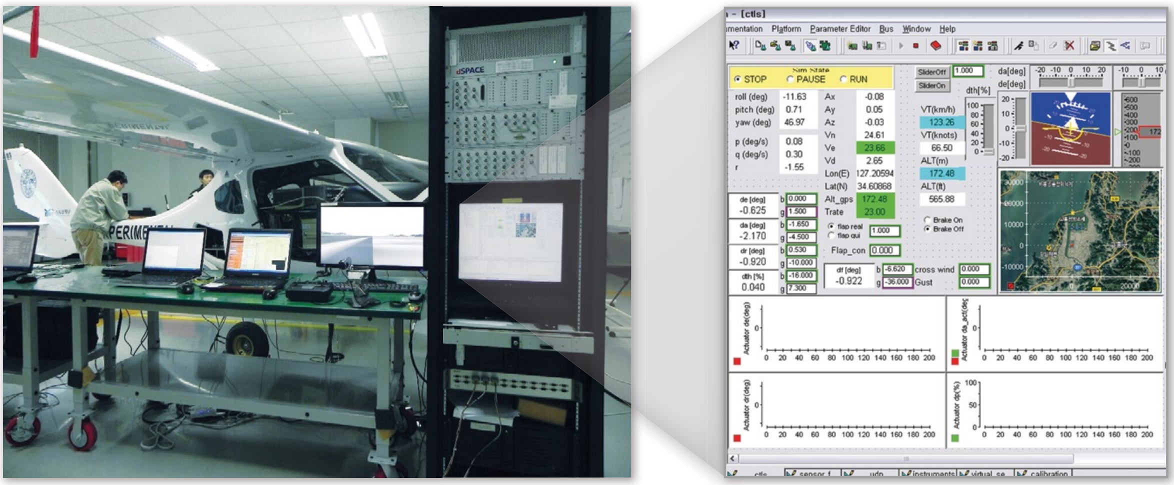
Virtual Test Flights in the Lab

The dSPACE system that simulates flights for flight control tests consists of a hardware-in-the-loop simulator with a DS1006 Processor Board that computes the flight maneuvers and associated sensor values. The simulator is connected to the flight control system onboard the aircraft via various I/O boards. It calculates the sensor values – the aircraft's position (GPS data), attitude relative to the direction of flight, acceleration, speed, etc. – and sends them to the flight control system via RS232. The flight control system uses this data to adjust the plane's control surfaces and guide it along the preplanned route. The positions of the control surfaces are then returned to the dSPACE Simulator to control the aircraft flight motion. All experiments are controlled and monitored from the test and experiment software dSPACE ControlDesk which supports tasks such as manipulating the experimental conditions (wind, etc.) and inserting any desired failures to test how the flight control system reacts – failed sensors or actuators and broken wires are some typical examples.
With this setup, comprehensive and complete tests can be performed without the aircraft having to leave the laboratory. This approach considerably reduces the number of real test flights while increasing the reliability of the overall system at the same time.

Unmanned Test Flights

The plane has already successfully performed its first test flight, which contained all the auto flight modes (table 1): stick auto, knob auto, loitering and point navigation. The flight control system automatically computes the resulting pilot stick positions to guide the plane. As a result of the accurate flight dynamics model and the preflight tests via the dSPACE HIL system, no tuning interventions were necessary.
With the model-based development process, the OPV's flight control algorithms can be developed much faster and at much less cost. The

Figure 2: The dSPACE Simulator executes virtual test flights in the laboratory. The test and experiment software dSPACE ControlDesk (shown on the right) is used to monitor all the experiments and insert failures.



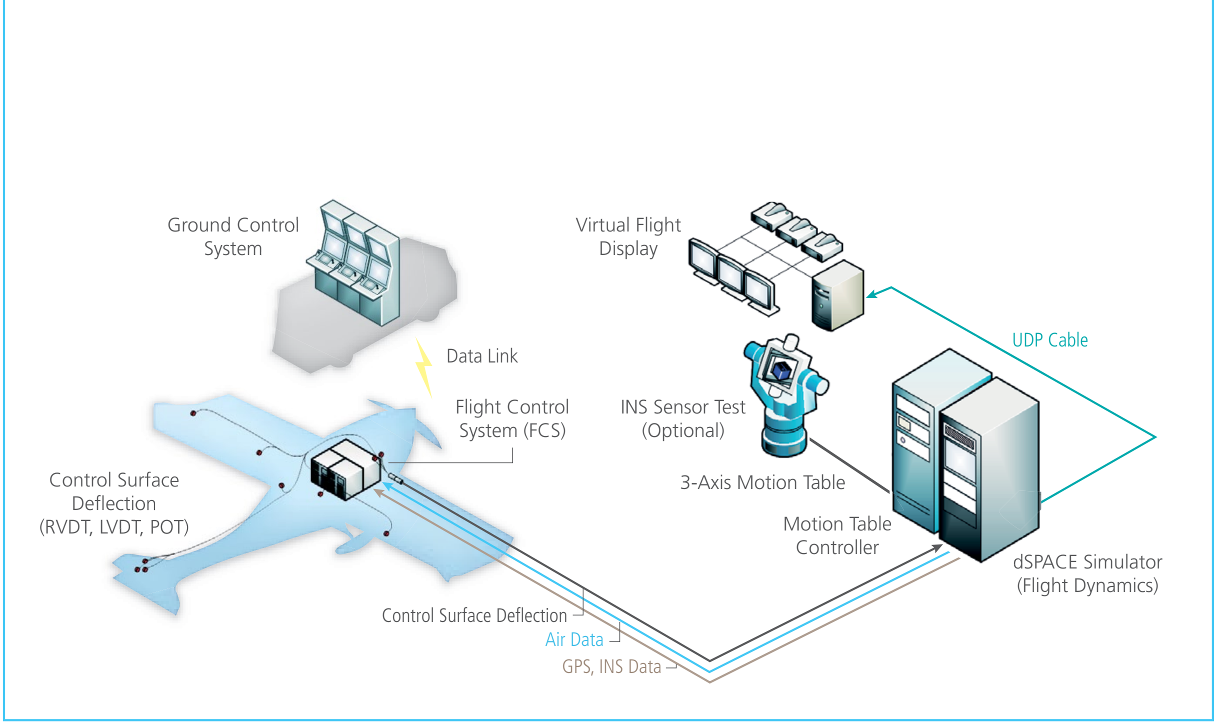


Figure 3: Schematic of the laboratory setup. A wide range of test scenarios can be performed on it without the plane having to leave the laboratory.

fact that the dSPACE tools are so well attuned to the MATLAB/Simulink world also considerably facilitates model-based development work. Being able to use function libraries is particularly helpful, as time is saved by reusing existing knowledge. ■

Dr. Hyoung-Sik Choi,
Sugchon Lee,
KARI

Stick auto mode	In this mode, the stick command is the attitude command so this is more stable than manual mode. As a result, the aircraft's attitude (pitch, roll, and yaw) follows the pilot stick command. In comparison, in manual mode the stick command is the control surface control command (for the elevator, aileron, rudder, etc.)
Knob auto mode	In this mode, the internal pilot gives a knob command (such as the altitude, airspeed, heading or roll) and the aircraft then follows the knob command.
Loitering mode	When the pilot engages loitering mode, the aircraft turns and makes a circle as long as loitering mode is engaged.
Point navigation (NAV) mode	When the pilot engages point NAV mode, the aircraft goes to the given target point. After passing the target point, the aircraft loiters as long as point NAV mode is engaged.

Table 1: Overview of the different auto flight modes that the OPV is able to perform.

Figure 4: The ground control station transmits commands to the aircraft and displays all the aircraft's instruments. Virtual 3-D graphics give a virtual cockpit view, which is useful for monitoring the aircraft's situation in case the pilot-view camera malfunctions.

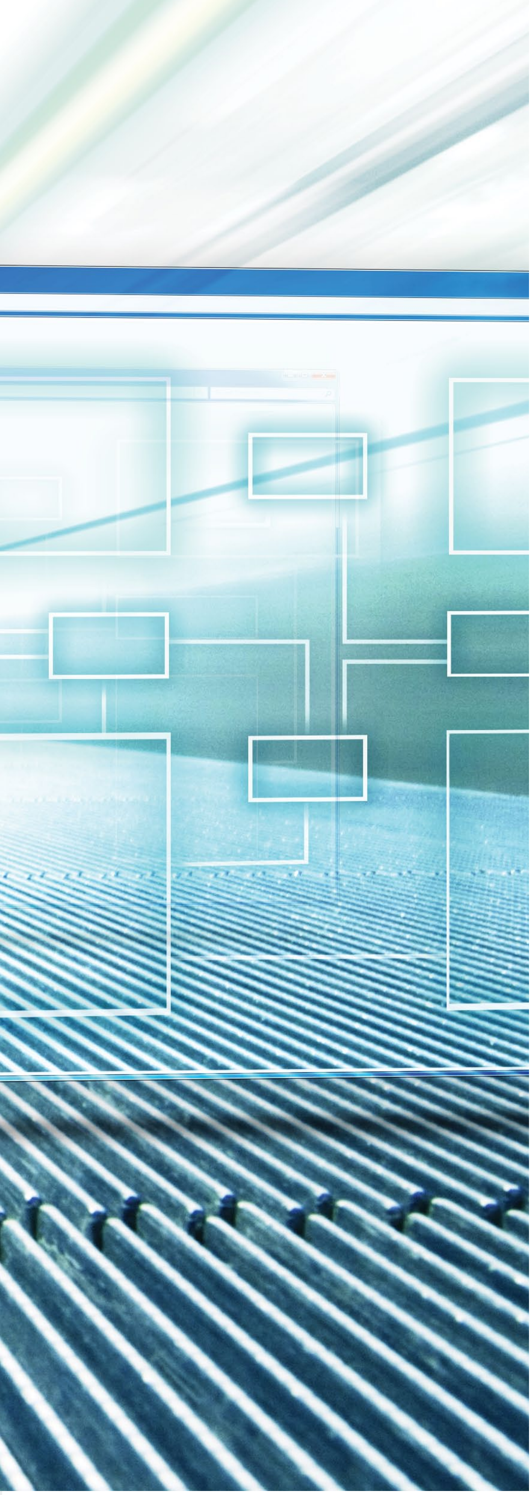
Dr. Hyoung-Sik Choi
Dr. Hyoung-Sik Choi is senior researcher in the Flight Control Team at KARI (Korea Aerospace Research Institute), Daejeon, Republic of Korea.

Sugchon Lee
Sugchon Lee is senior researcher in the Flight Control Team at KARI (Korea Aerospace Research Institute), Daejeon, Republic of Korea.

Industrialized integration and validation of TargetLink models for series production

Continuous Software Production

The complexity of software systems in vehicles is increasing at the expense of the development of new innovative driver assistance functions. The EB Automotive Software Factory is the key to success because it implements recurring work processes as a production line for software. This means that integration and validation steps can be reproduced and traced at any time, in AUTOSAR-compliant or ISO 26262-compliant projects and also in conventional projects.



Increasing Complexity

When new technologies are integrated in vehicles, more of these technologies involve the actual software. The key task is to connect the growing number of individual systems from different suppliers with each other. The network of actuators and sensors is expanding. There are new application scenarios and the proportion of safety-related functions is also increasing. Complying with the required standards, such as ISO 26262 for the functional safety of road vehicles, is raising the quality requirements even more. The reality for premium vehicle manufacturers today is a variety of about 60 software features, consisting of several hundred software modules on around a dozen electronic control unit (ECU) platforms that are produced by a handful of suppliers and used in 25 series. This increases the complexity during development and during the management of new projects at a disproportionate rate. And today's development teams cannot just be re-scaled to meet this increase. Thus, a wide variety of challenges arise when innovations must be brought up to series quickly and successfully.

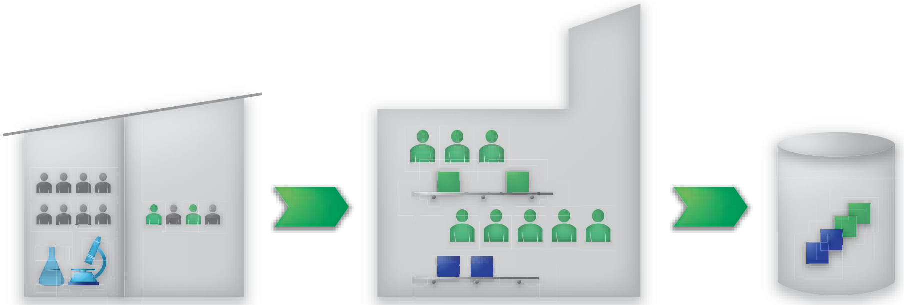
Software as a "Component"

Due to this complexity, in 2010 a German premium car manufacturer brought Elektrobit on board as a partner for developing the software of driver assistance systems. The goal was to let the car manufacturer's own development team focus on its core competencies, namely, developing innovative driver assistance functions in MATLAB®/Simulink®.

The software development experts at Elektrobit took on the task of implementing the functions in TargetLink® models on various ECUs and in several series and variants, integrating them with Tier 1 platforms and validating functional safety. The business model that was agreed on handles the software products that Elektrobit integrated like "components". The products are given their own identification number and delivered to production lines all over the world. To take full advantage of the potential that this business model offers, Elektrobit set up the EB Automotive Software Factory for model-based tool chains, in analogy to vehicle production lines. This process generates executable code from models, checks to see if guidelines are being met, and tests the software.

A Virtual Project House

To operate the Software Factory and to be able to independently perform all the tasks being done at the EB locations, EB established a virtual project house and a stable network connection, according to the customer guidelines, to exchange deliveries and balance the project management systems. Elektrobit provide these specific development services for their customers' project teams in a private cloud.



Customer Research & Development

- Product Vision
- Architecture, Interfaces
- Design
- Algorithms
- Planning

EB Automotive Software Factory

- Incoming Goods Control
- Production and Integration
- Test and Verification
- Delivery
- Controlling and Reporting

Software Product

- Ready for SOP
- Documentation
- Release Notes
- Traceability
- Verified Quality

© Elektrobit Automotive GmbH

The production lines in the EB Automotive Software Factory.

EB Automotive Software Factory – The Big Picture

The Software Factory is a coordinated multi-step workflow made of modular build steps that ensure continuous integration and delivery. The workflow is used for different types of releases. For example, intermediate results are generated for each modification to give developers feedback as quickly as possible on whether the modifications can still be integrated. The production releases are based on a selected software configuration, started manually and released.
The following measures enhance this concept of continuous delivery:

- Testing the delivered components to verify that they match the software quality requirements and to ensure their integratability
- Continuous monitoring of the metrics (for software quality, requirements or test coverage, number of issues, etc.)
- Continuous monitoring of the Software Factory itself (CPU utilization, length of build queues, etc.)
- Traceability, such as between the release and the version of the integrated software
- Automatic generation of a complete documentation
- Automatic provisioning of build machines to scale the computing capacity as desired

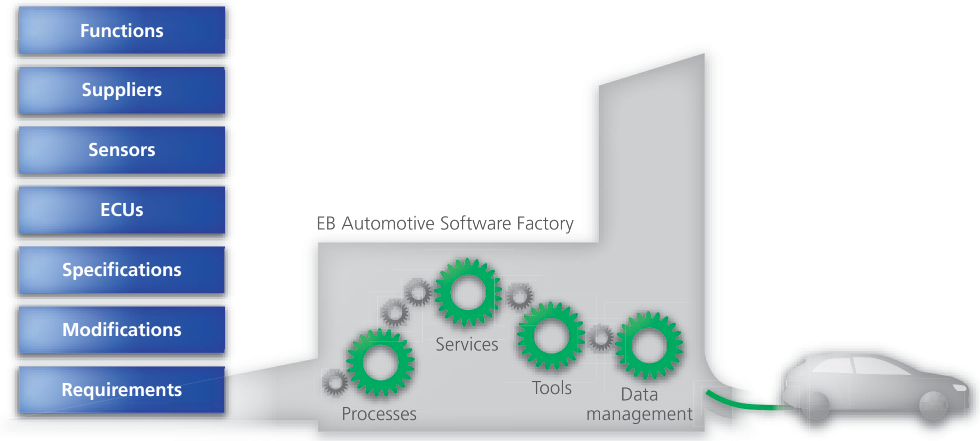
Essential Requirements on the Factory

There are some features that are essential for setting up a Software Factory and integrating complex systems:

- The software architecture must reduce integration costs and enable meaningful and efficient tests
- Configuration management must deliver consistent versions from build environments, software, tests, etc.
- Variant management must reduce the number of versions to be integrated and verified
- Practical error prevention methods must reduce the effort for error detection

During the starting phase, the manufacturer's/OEM's tool and process landscape was analyzed to see which elements can be used and to create the interfaces necessary for a comprehensive, homogeneous software development process.

> "In our software development process for safety-related functions according to ISO 26262, we rely on dSPACE's production code generator TargetLink."
>
> *Robert Holzwarth, Elektrobit*



© Elektrobit Automotive GmbH

The concept of the EB Automotive Software Factory.

Tool Landscape

The resulting tool chain consists of the following components:

- MATLAB/Simulink/Stateflow®, with other toolboxes
- dSPACE TargetLink as the code generator
- MES Model Examiner® for static model verification
- Polyspace® for static code analysis
- MTest and TPT for model module tests
- Mini HIL systems and tools from the vehicle manufacturer/OEM
- Jenkins as the continuous integration system
- PTC Integrity for configuration, change and release management at the OEM
- Subversion and JIRA for version and change management at Elektrobit
- pure::variants for variant management
- Further scripts and tools for
    - Model integration and system integration
    - Variant management
    - Tier-1-specific adaptations
    - Generating AUTOSAR interfaces
    - Interface tests

Mapping on Virtual Machines

The actual build environment was mapped to virtual machines (VMs) to keep the installation and especially the configuration of the tools under control.

Different versions and variants, such as MATLAB R2007 and TargetLink 3.0 or MATLAB 2011b and the 64-bit TargetLink 3.4, were mapped on their own instances. The central continuous integration systems are based on these VMs. Local instances are provided as the development environment for several applications. For example, Factory 2 Go is used to generate software versions in a consistent build environment during test drives without any online connections.

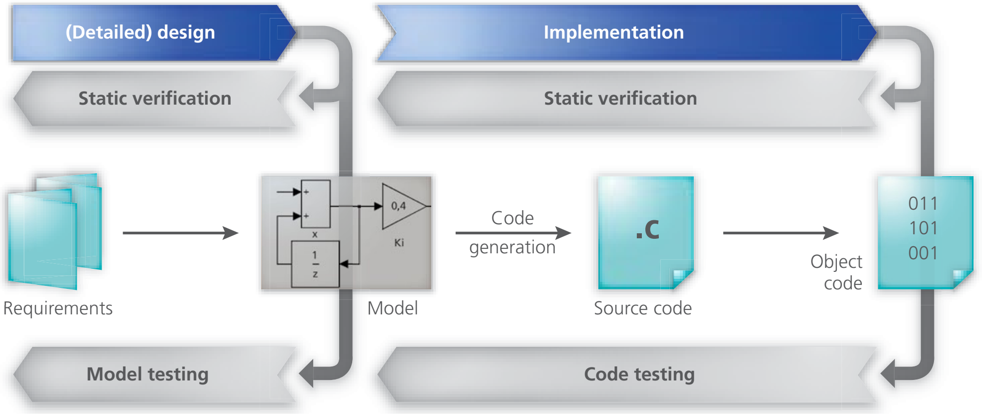
The advantages of virtual machines are also used for the long-term archiving of complete projects, since only files need to be archived, not any hardware.

Build Process

The designed process is based on automating and modularizing each step with clear input and output artifacts. For example, at the interface between the OEM and Elektrobit, there is an automatic test of incoming goods set up for MATLAB/Simulink models and C code. This test involves the following tasks:

- measure the complexity of the modules
- check whether the relevant modeling rules and design guidelines are met
- check the consistency of the internal and external interfaces with respect to the specified design
- generate reports on the findings and metrics

The model integration step requires the models and components and the variant model of the OEM as the input artifacts. The model modules specified in the variants are used in a generated interface framework within the overall model and connected or replaced by empty modules. Code generation with TargetLink is fully automatic. During post-processing, hook functions are used to perform Tier-1-specific modifications in the resulting C code. Measurement and parameter files are



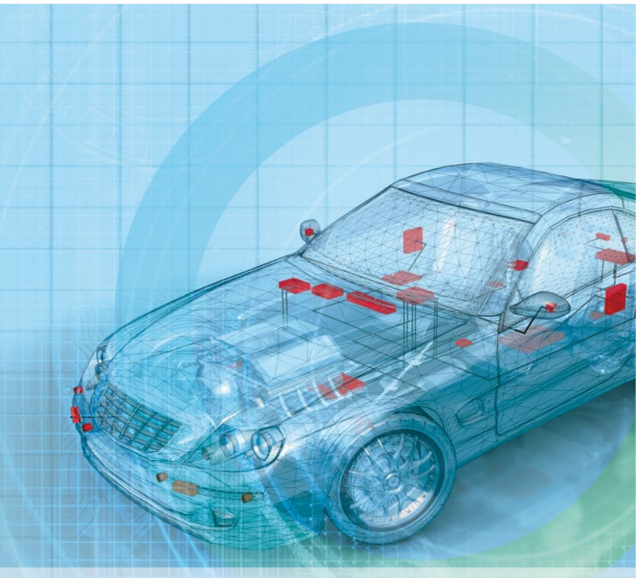
ISO 26262-6, 6.9 Software unit testing
ISO 26262-6, 6.10 Software integration and testing

The TargetLink reference workflow provides guidance on how to fulfill functional safety requirements with model-based development methods and tools.

generated, and the layout of the generated header files is controlled by style sheets and parameterized by the TargetLink Data Dictionary. The tool created by the vehicle manufacturer/OEM for AUTOSAR frame generation and post-processing was simplified considerably in cooperation with dSPACE and reflected in the new features of TargetLink 3.4.

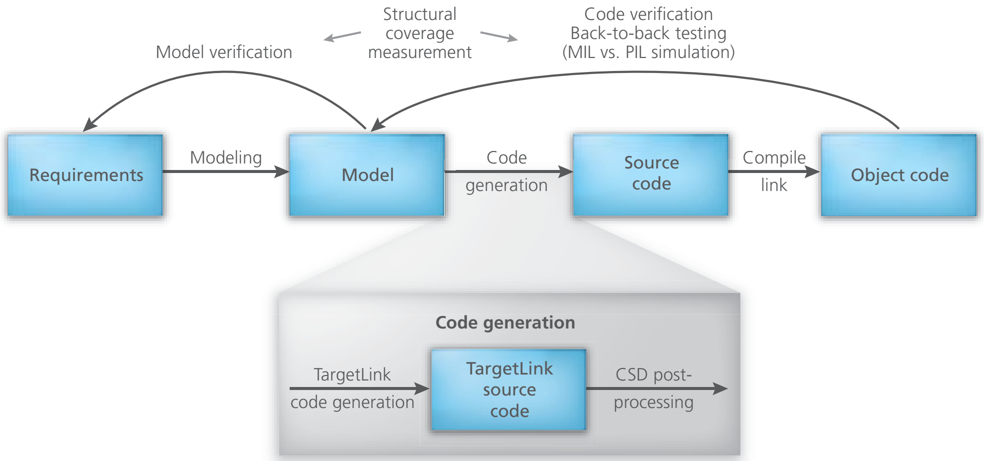
The TargetLink Reference Workflow

ISO 26262 requires that the use of software tools for developing safety-related software meet the requirements of Part 8, Section 11. This means that the tool's reliability must be evaluated in the concrete project environment and that the tool must be given a quality level that reflects the result. The production code generator TargetLink itself was certified by TÜV SÜD for the development of safety-related systems. The certificate confirms that TargetLink is suitable for software development according to ISO 26262 up to ASIL-D, IEC 61508 up to SIL 3, and the derivative standards. dSPACE also provides a reference workflow document on the model-based software development of safety-related systems with TargetLink. This document was also examined by TÜV SÜD and acts as a guideline on how to meet the specific safety standard requirements in safety-critical projects. Thus, when the reference workflow is used, no further qualification measures are required for TargetLink. For the self-developed post-processing procedure, the TargetLink Reference Workflow was examined in



Conclusion

Elektrobit successfully took on the production of production software for several product lines and generations at an OEM. "Software components" are created in a customer-specific private cloud and delivered to the production teams. The continuous integration of the software is enhanced by measures to ensure process and software quality. Elektrobit was able to use the TargetLink Reference Workflow successfully and expand it on their own in-house tools. Elektrobit's experience and expertise in their EB Automotive Software Factory are key factors that help manufacturers be successful when they perform complex projects to develop innovative software for series production. By auto-



How the reference workflow enhances validation.

> “The production code generator TargetLink makes complex processes such as AUTOSAR-compliant code generation extremely easier.”

Robert Holzwarth, Elektrobit

more detail to evaluate the process step that was downstreamed by Elektrobit. The evaluation showed that the recommended measures, such as back-to-back tests, also cover this downstreamed process step. This means that here too, no further tool qualification measures are required when the TargetLink Reference Workflow is used. dSPACE and Elektrobit work together hand in hand to implement the reference workflow and especially to perform the recommended back-to-back tests. ■

Robert Holzwarth,
Elektrobit (EB) Automotive GmbH

mating software production and delivery, Elektrobit has delivered more than 230 releases to its customer on time since 2010, with over 75 individual production releases. Elektrobit continues to expand the concepts of the EB Automotive Software Factory. For example, they are making the supply chain processes more transparent and thus easier to model. The automotive, commercial vehicle, and component supplier branches can access project-specific implementations of the Software Factory worldwide in a private cloud.

Robert Holzwarth
Robert Holzwarth is Head of Technology & Innovation Software Integration and Services at Elektrobit Automotive in Erlangen, Germany.

Developing a Self-Driving Vehicle

No Driver Required

Thanks to advances in the development of intelligent drive technology, the time is coming closer when self-driving vehicles will be a part of everyday road traffic. With the help of dSPACE MicroAutoBox, a group of researchers at School of Automotive Studies, Tongji University, China, developed an electric prototype vehicle that can already drive by itself on the university campus.

A vision that is closer to becoming a reality:
Self-driving vehicles in everyday traffic

Vehicle Control via MicroAutoBox

The electric prototype vehicle developed by the research team at Tongji University is based on four wheel hub motors with a 140-volt lithium battery. The vehicle handles all sorts of driver assistance functions, including lane keeping and lane changing, adaptive cruise control (ACC), emergency braking, stopping at a stop line, and merging into the traffic flow. MATLAB®/ Simulink® is used to develop the functions of the subsystems and two MicroAutoBoxes are used as the control centers of the vehicle.

Combining Several Environment Sensors

To detect the surroundings in detail, the prototype vehicle uses four different types of sensors: namely, camera, GPS, laser radar (lidar), and millimeter wave radar.

The camera and GPS are used to identify the road. To recognize the roadway, the camera detects the side lines of the road. If the road does not have any side line markings, the lane can also be generated with the help of GPS. Lidar and millimeter wave radar are used to identify the relative positions and relative speeds between the vehicle and obstacles or other vehicles. This information serves as the essential basis for a number of driver assistance functions such as adaptive cruise control (ACC), where the vehicle always maintains a safe distance to the vehicle ahead.

The vehicle prototype designed by Tongji University can already drive on its own around the university campus.

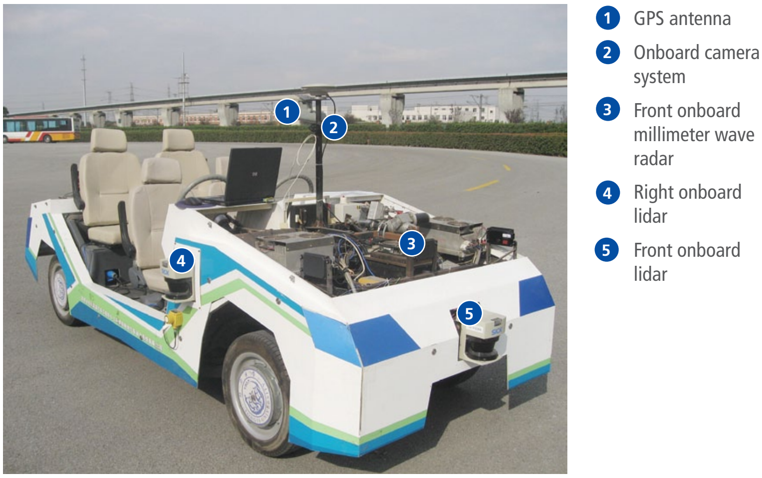


Figure 1: For orientation, the prototype vehicle uses GPS, camera, laser radar (lidar) and a millimeter wave radar.

Two-Stage Path Planning

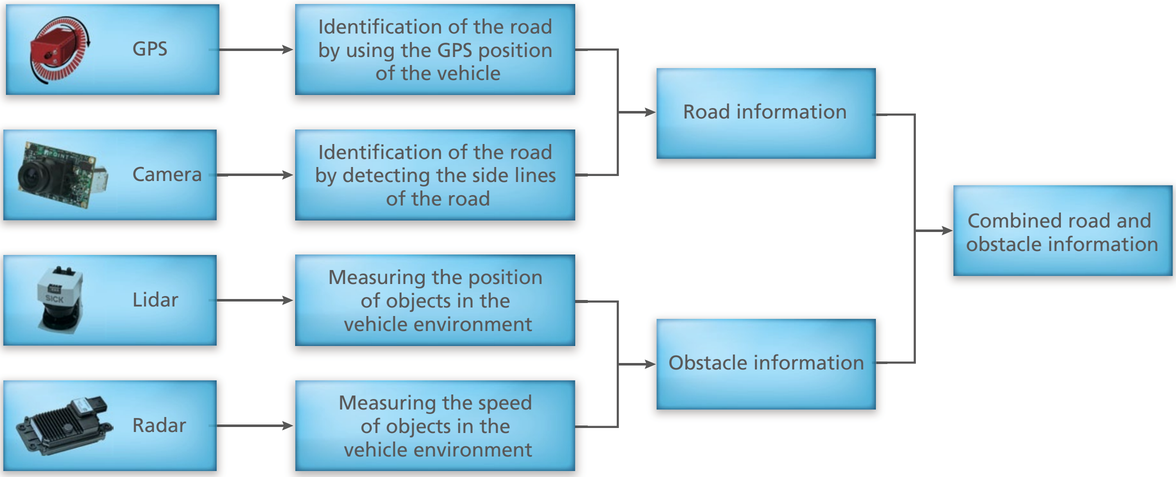
The route is planned in two stages: The first stage is global path planning. This involves using a digital map and the vehicle's current position data to calculate the shortest route from start to finish. The second stage is local path planning. This involves dividing the previously planned global route into short segments. The segments are calculated by cubic spline functions in such a way that the ends of the splines flow into one another. This is used to keep the steering system from suddenly jerking at the interpolation points of the segments. The local path planning outputs the actual steering angle and the speed commands for the vehicle.

Two MicroAutoBoxes in Use

The control algorithms were developed entirely in MATLAB®/Simulink®, so using dSPACE development tools was a natural choice because they are an optimal fit for the MATLAB/Simulink development environment. A particular advantage of the MicroAutoBox is its compact and robust design, which makes it ideal for use in prototype vehicles.
The model, which was designed in Simulink, is automatically coded and implemented on the MicroAutoBox. And the large number of interfaces and driver modules provided by MicroAutoBox simplify every implementation work step. Installation of driver modules in the Simulink model via drag & drop is just one typical example of this convenient work method. On the whole, the dSPACE development environment significantly simplifies several steps, which saves a great deal of development time. The first of the two MicroAutoBoxes collects all the im-

> "Thanks to its robustness and easy configurability, MicroAutoBox is the optimal tool to use in prototype vehicles."
>
> *Prof. Hui Chen, Tongji University*

Figure 2: The road is identified by the camera and/or GPS. Lidar and radar sensors give the vehicle information on the surrounding traffic.



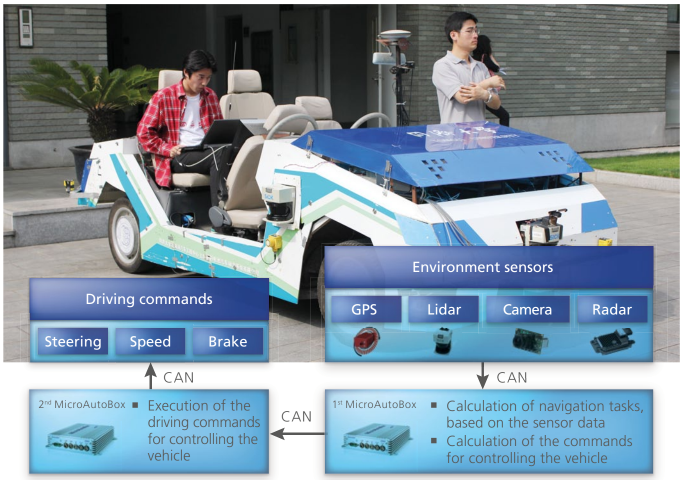


Figure 3: Two MicroAutoBoxes are used in the vehicle. The first MicroAutoBox evaluates the sensor data and calculates the navigation data. The second MicroAutoBox is used to actually control the vehicle (steering, braking, etc.).

portant navigation-relevant data that the various sensors deliver regarding the vehicle environment and uses this data to calculate the commands necessary for controlling the vehicle. The second MicroAutoBox receives these commands via CAN bus and then performs the actual vehicle control (steering, braking, etc.). If necessary, the driver can still take over control of the vehicle at any time by grabbing the steering wheel. In addition, the motor can be switched off automatically for safety reasons.

Autonomous Trips on Campus

The test vehicle is capable of steering around obstacles, such as a pedestrian or another vehicle, while driving on the test route around campus. With the help of a digital map, the vehicle also safely masters stopping at stop lines and cornering. If another vehicle is driving ahead slowly, the test vehicle is also able to follow automatically at a safe distance.

More Sensors in the Future

The next stages of the project will focus on the vehicle's environment detection system because all kinds of autonomous driving technologies are based on this. Moreover, as sensor technology becomes more advanced, the spectrum of automotive sensors that can be used at an acceptable price will also increase. Therefore, future research activities will focus on combining the measurement data from the various sensors and increasing the fault tolerance of the vehicle control. The dSPACE development environment will also be used for these future stages. ▪

Prof. Hui Chen,
Tongji University

> "With MicroAutoBox, we can experience new driver assistance functions in the test vehicle immediately."
>
> *Prof. Hui Chen, Tongji University*

Summary

The electric vehicle prototype developed by Tongji University handles various driver assistance functions such as adaptive cruise control, automatic lane keeping and lane changing, stopping at stop lines, and emergency braking. Two MicroAutoBoxes are used in the vehicle. The first MicroAutoBox evaluates the data from the environment sensors (GPS, lidar, radar, camera) and plans the route, the second MicroAutoBox acts as the driver and controls the vehicle.
This makes the vehicle able to drive autonomously on the university campus.

Prof. Hui Chen
Prof. Hui Chen is Director of the Chassis Electronic Control Systems Lab at the School of Automotive Studies, Tongji University in Shanghai, P.R. China.

Optimized Hybrids

Researchers from the University of Valenciennes and Hainaut-Cambrésis are developing new energy management algorithms to improve the fuel economy of micro and mild hybrid vehicles. dSPACE MicroAutoBox and further modular dSPACE systems are used to implement the control strategies both on test benches and in vehicles.

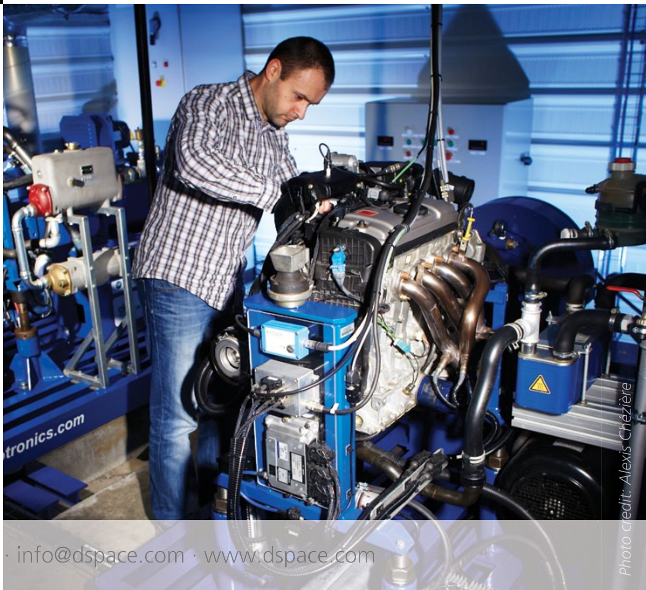
How optimized algorithms in an ECU save fuel

Photo credit: Alexis Chézière

Today, hybrids are on the road, from affordable micro hybrids providing mostly the stop&start feature, up to more expensive systems with significant pure electric mode range. One of the challenges is to improve the control algorithms to reach better fuel economy and reduce emissions. Developing efficient energy management algorithms (EMAs) does slightly increase the hybrid system costs, but it can also improve the vehicle's fuel economy by several percent. That has been precisely one of the research topics of the Laboratory of Industrial and Human Automation Control, Mechanical Engineering and Computer Science (LAMIH UMR CNRS 8201) since 1996.

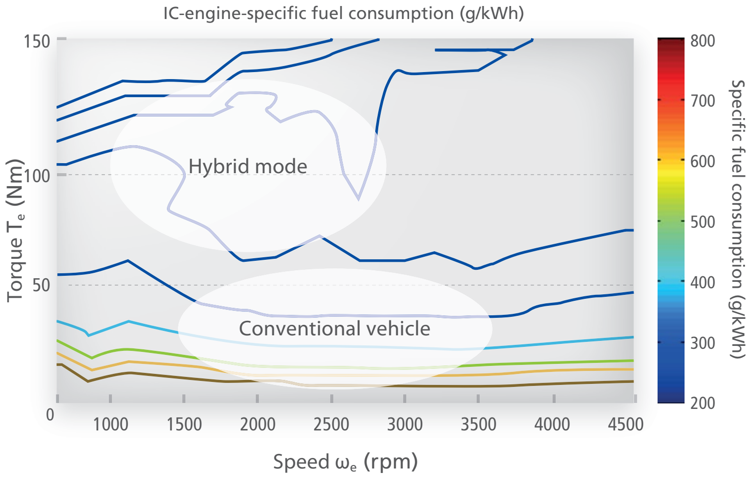
Energy Management Algorithms

The first functionality to be controlled by the EMA is the engine stop&start. As soon as the vehicle is going to stop, the internal combustion (IC) engine is stopped. When the driver steps on the pedal or selects a gear, the IC engine starts up again. This is one of the major fuel economy sources during urban driving conditions. The EMAs also compute the amount of regenera-



New hybrid test bench to develop energy management algorithms in the lab.

Photo credit: Alexis Chézière



The fuel consumption map provides the specific fuel consumption for all speed-torque combinations of an internal combustion (IC) engine. The optimal engine operating point shifts in hybrid drive mode .

tive braking according to the energy storage level. The basic idea is to recover as much energy as possible (recuperation) without compromising the driving comfort. The driver should not perceive the additional deceleration induced by the electric machine.

The Challenge: Hybrid Driving

The most difficult mode to be controlled is hybrid driving. In this phase, the electric machine and the engine are used together to improve the overall powertrain efficiency. Basically the IC engine load is increased and the resulting additional mechanical power is used by the electric machine to recharge the energy storage while driving. This is done until the energy storage is full. Then either the IC engine load is decreased, but this also decreases the efficiency, or, if available, the pure electric mode can be used. In any case, mathematical optimization is required to ensure good powertrain efficiency.

Evaluating Mild Hybrid Vehicles

Hybridization systems with low rated power and small-capacity energy storage systems are less expensive and thus more competitive on the market. These systems provide mostly the stop&start feature along with regenerative braking. Pure electric mode can also be used in some particular situations (low speed, low power request). The BELHYSYMA (Belt Hybrid System Management) project objective was to evaluate the potential of a mild hybrid system. To demonstrate the importance of the EMA, several control laws were assessed.

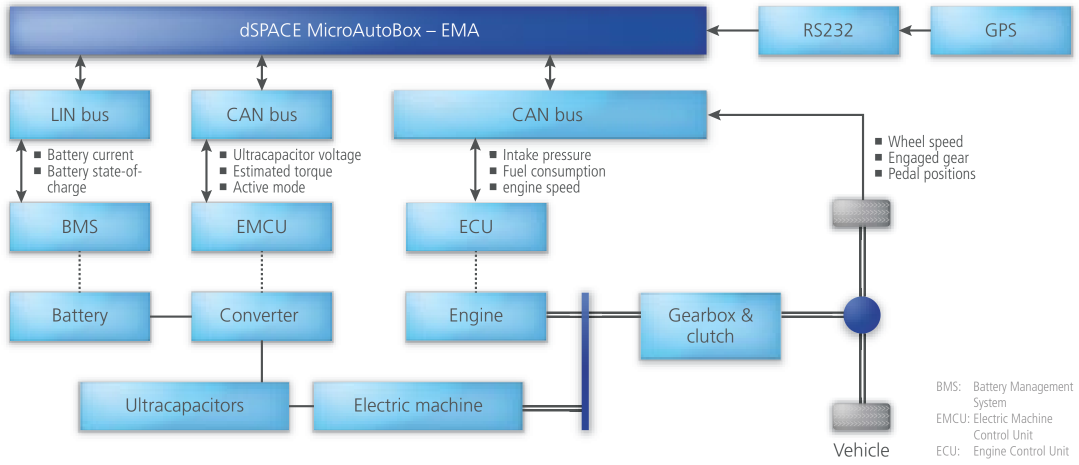
MicroAutoBox as a Top-Level Processing Unit

Integrating a mild hybrid system into an existing vehicle is a great challenge and requires a wide range of I/O interfaces (LIN, CAN, RS232, and analog I/O). dSPACE MicroAutoBox was chosen as the central processing unit since it provides connectivity for most of the sensor interfaces (analog voltage, LIN, CAN, RS232, etc.) and, if needed, RapidPro units can be used to connect the MicroAutoBox to additional sensors and/or actuators.

Using dSPACE software tools, the GPS data of the NMEA-0183 protocol (a standard for communication between navigation devices) can be easily decoded and integrated into our applications. The RTI CAN toolbox allows importing CAN DBC files (database CAN, a file format for exchanging CAN base data) with just one click, for fast and accurate connection of the MicroAutoBox to the other vehicle control units (engine, electric machine, etc.). The battery is monitored by a specific sensor connected through the LIN port.

Thanks to all the different sensors (analog voltage, LIN, CAN, RS232, etc.) and a set of test bench experiments, a detailed model of the IC engine and the whole vehicle has been developed and calibrated. During urban driving conditions (ARTEMIS driving cycle), the hybrid system and a classical EMA approach allow improving fuel economy by 9.5% (compared with a conventional car). This result is strongly dependent on

> “We have chosen the dSPACE MicroAutoBox and its associated RapidPro extension in order to get a wide range of connectivity. Whatever our future needs, we know that we will have the proper interface with our control system.”
>
> *Sébastien Delprat, University of Valenciennes and Hainaut-Cambrésis*



Setup of a prototype mild hybrid system. The energy management algorithms are implemented on the MicroAutoBox.

the stop&start strategy, which is mostly tuned according to the expected driving comfort and number of stop&start operations. By using the proposed EMA, fuel economy is raised to 14.1%. So, by optimization resulting in a simple modification of the control software, fuel economy is improved by another 4.6%.

Next Steps Toward Better Energy Performance

Improving micro hybrid vehicle performance through control strategy design is a real challenge due to the limited capacity of the hybrid system. Nevertheless some interesting results have been demonstrated on the prototype. To improve the experimental analysis, a hybrid test bench was designed, so the EMA can be tested without vehicle or human interaction for more reproducible tests. It is composed of an IC engine and a mild hybrid system coupled to an eddy brake and a motor to emulate the regenerative braking phases. It is controlled by a dSPACE system mounted in PX20 Expansion Boxes. ■

Sébastien Delprat,
Clément Fontaine,
Sébastien Paganelli,
University of Valenciennes and Hainaut-Cambrésis

Acknowledgement
The BELHYSYMA project is funded by the French Nord Pas de Calais region, the European Funds FEDER and the French DGCIS.

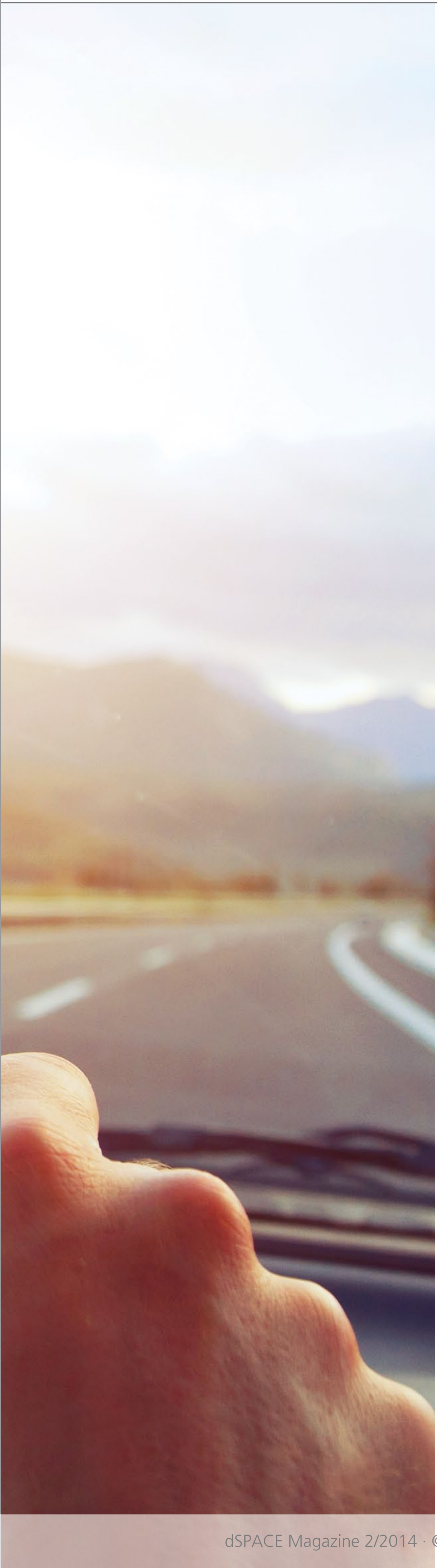

Photo credit: Alexis Chezière

Sébastien Delprat
Sébastien Delprat is Professor at the University of Valenciennes and Hainaut-Cambrésis, France.

Sébastien Paganelli
Sébastien Paganelli is Project Engineer at the University of Valenciennes and Hainaut-Cambrésis, France.

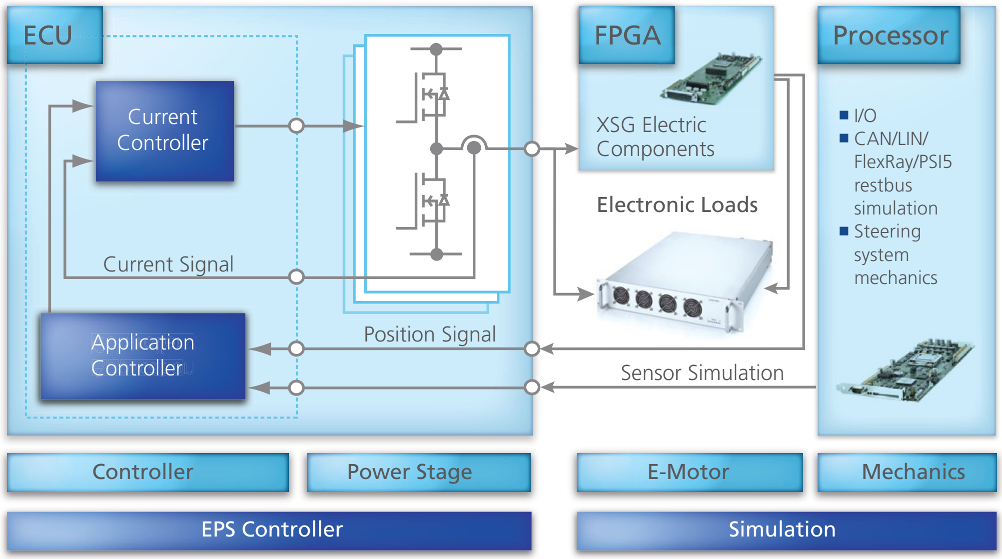
Safe steering

Test system for validating the ECU of an electric power steering system at power level



The electric power steering system is among the more safety-critical components of a vehicle. Its development must therefore meet stringent safety requirements. Japanese steering system manufacturer JTEKT has installed an ISO-26262-compliant process and a powerful test system for ECU development.

The test system for steering system ECUs with an operator workspace.



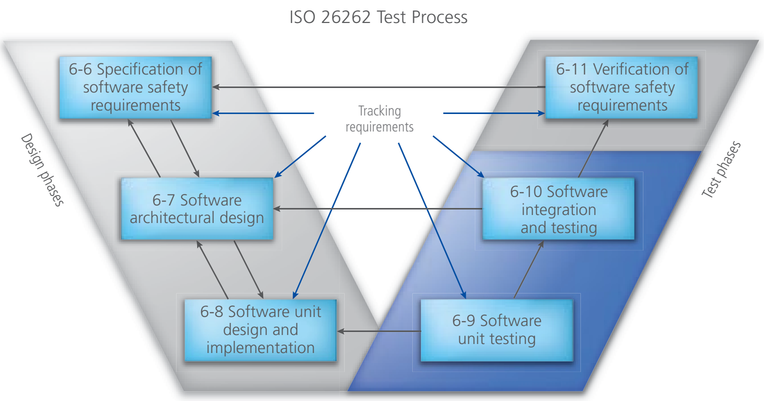
Load simulation components. The electric power steering (EPS) ECU is displayed on the left-hand side, the simulator including the processing units (FPGA, processor) plus the load module on the right-hand side.

Developing Safety-Critical Systems

JTEKT supplies automobile manufacturers with components such as steering systems and driveline components which have to meet strict safety requirements.

This is particularly true for electric power steering (EPS) systems, which have to be developed according to ISO 26262, the functional safety standard for road vehicles. It defines functional safety for automotive equipment applicable throughout the life cycle of all automotive electronic and electrical safety-related systems. A basic requirement for ISO-26262-compliant development is a complete, seamless development process that includes customer specifications, evaluation of interim results, and final verification.

The software development process for the electric power steering system. The test activities cover the areas of software unit testing and software integration testing.



Safety Requirements for Electric Power Steering Systems

At the beginning of a development project, a hazard analysis and risk evaluation is carried out. Applicable Automotive Safety Integrity Level (ASIL) requirements are identified and allocated. High-level functions of EPS systems are generally classified as ASIL D, the safety integrity level with the strictest requirements. There are more than 100 test requirements (items) for validating electronic control units (ECUs) according to ISO 26262. A clearly defined and structured work process ensuring complete testing is needed to efficiently meet these requirements in all customer projects. In particular, the requirements have to be traceable all the way to the test results.

Solution: A Seamless Tool Chain

A seamless, transparent process requires an appropriate tool chain.

> “In order to efficiently evaluate the controller of our electric power steering system according to ASIL D safety requirements, automatic test execution with a hardware-in-the-loop (HIL) simulator is indispensable.”
>
> *Hirozumi Eki, JTEKT*

The core components of such a tool chain are requirements management and an ECU test system. For requirements management, JTEKT uses the IBM® software DOORS®. For ECU testing there are many established methods. Hardware-in-the-loop (HIL) simulation is used if automated, reproducible tests are required, and JTEKT uses associated simulators. To validate new families of electric power steering (EPS) systems, a HIL system setup tailored to the test task was to be designed.

Basic Requirements for the Test System

EPS ECU testing must be performed at power level, since the ECU cannot be modified to bypass the power stage. One of the main tasks of the HIL simulator is therefore the emulation of the electric motor. Motor currents have to be emulated precisely. The HIL simulator must therefore be able to exactly simulate fast processes and high currents. But the simulator also has to be flexible enough to simulate all automotive components that are relevant for ECU testing. All tests have to be automated, and requirements, test cases, and test results must be linked seamlessly and must be traceable.

Project-Specific Requirements

The detailed requirements for developing an electric power steering system comprise technical and organizational aspects:

1) Creating automated, ready-to-use test sequences for test engineers

2) Generating the required HIL signals for the ECU as specified in test sequences by the user

3) Measuring signals of the HIL simulator and the ECU, and adjusting parameters during the test sequences

4) Automatic evaluation of test results as status reports

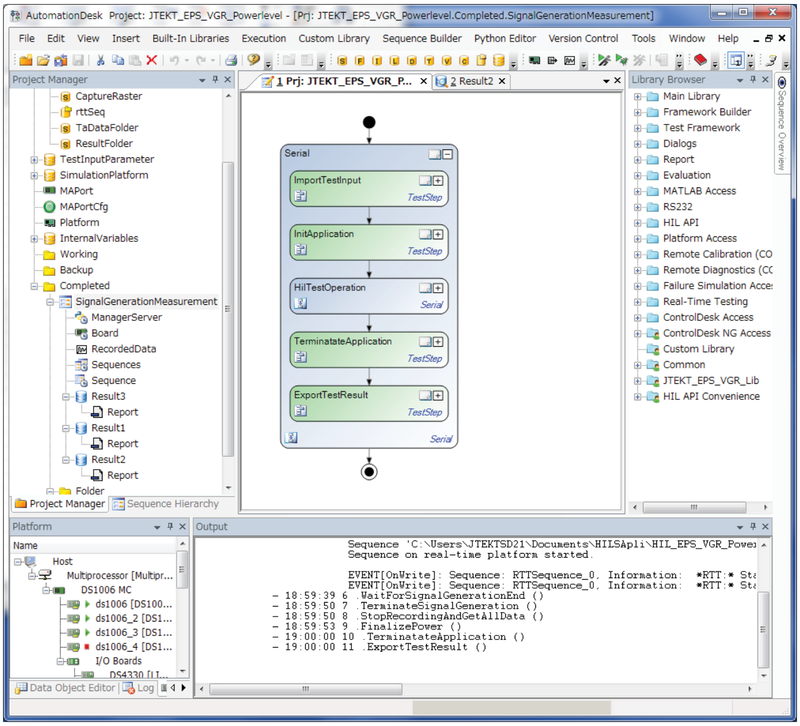
5) Exporting test data (measuring data from (3), evaluation from (4)) for an external system

6) Generating motor simulation values (motor phase current, motor phase voltage at ECU power stage, and back EMF)

The Test System

After JTEKT studied and assessed different solutions on the market, dSPACE was commissioned to build a test system according to the requirements of the EPS project. The simulator consists of a processing unit (a quad-core DS1006 Processor Board) to simulate mechanical automotive components and various tasks at the restbus, as well as a fast FPGA (field programmable gate array) processing platform (DS5203) to calculate the electric motor. The FPGA controls an electronic load system to emulate the electric motor. For this, the Electronic Load Module DS5381 is used. It has cascade-switching MOSFET power stages that reach switching frequencies of up to 3.2 MHz, thereby

dSPACE Simulator with the highly dynamic load emulator DS5381 (center).



```
                Sequence 'C:\Users\JTEKTSD21\Documents\HILSApli\HIL_EPS_VGR_Powe
                Sequence on real-time platform started.

                EVENT[OnWrite]: Sequence: RTTSequence_0, Information:  *RTT:* Sta
                EVENT[OnWrite]: Sequence: RTTSequence_0, Information:  *RTT:* Sta
        - 18:59:39 6 .WaitForSignalGenerationEnd ()
        - 18:59:50 7 .TerminateSignalGeneration ()
        - 18:59:50 8 .StopRecordingAndGetAllData ()
        - 18:59:53 9 .FinalizePower ()
        - 19:00:00 10 .TerminatateApplication ()
        - 19:00:00 11 .ExportTestResult ()
```

AutomationDesk defines, performs, and evaluates the automated tests.

ensuring a highly dynamic simulation of the electric motors. The simulator also has CAN/LIN/FlexRay interfaces to perform a restbus simulation of the connected systems. A peripheral sensor interface 5 (PSI5) board (DS2302) is available for the simulation of sensors of safety-critical applications. A Failure Insertion Unit (FIU) equipped with high-current contactors is included to test ECU behavior in case of a short-circuit to another wire or ground. The electric motor simulation of the brushless DC (BLDC) motor of the EPS is performed on the FPGA using a BLDC motor model from the XSG Electric Components library. Models for the position encoder and the angular processing unit were implemented from the same library.

For signal evaluation, a software scope from the XSG Utils library is included on the FPGA. It can be used to record and display simulated signals (such as motor currents), or measured FPGA IO signals (such as phase voltages of the ECU power stage). The temporal resolution is in the range of nanoseconds.

Evaluation of the Test System

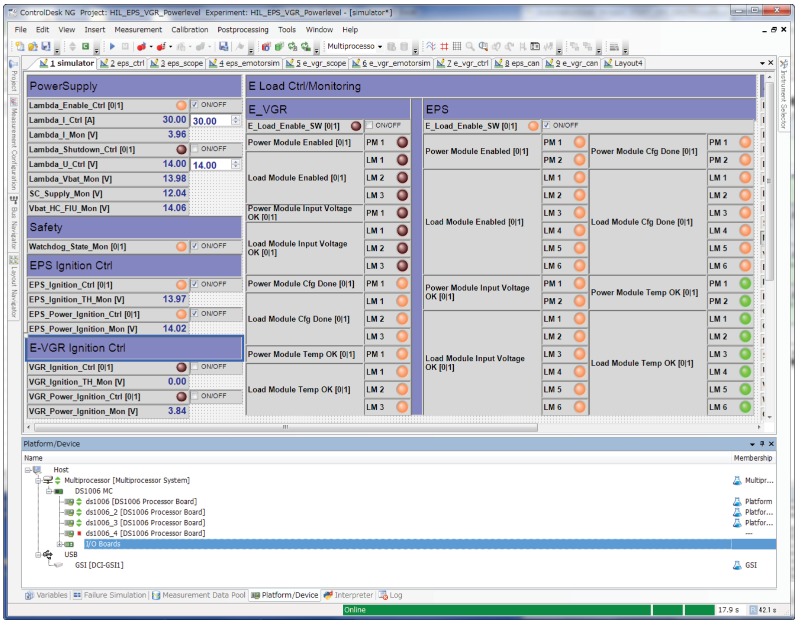
When the simulator was delivered, JTEKT was able to start operation immediately with the help of dSPACE Engineering. No changes or further settings were needed, saving the developers the entire preparation time. dSPACE even continues their support after the introduction of the HIL simulator.

The installed test system meets the requirements for a comprehensive EPS test. The evaluation of measured and simulated currents, torques, and position signals for different steering maneuvers delivers plausible results with the expected precision. The implemented FPGA processing platform in combination with the load unit ensures a stable closed-loop operation of the EPS ECU and the simulated electric motor with suitably fast sample times.

The combination of ControlDesk Next Generation with AutomationDesk makes the test system easy to use and automatable.

In addition, the DCI-GSI1 (generic

Signal monitoring and display in ControlDesk Next Generation.



PowerSupply

Lambda_Enable_Ctrl [0|1]	●	☑ ON/OFF
Lambda_I_Ctrl [A]	30.00	30.00
Lambda_I_Mon [V]	3.96	
Lambda_Shutdown_Ctrl [0|1]	●	☐ ON/OFF
Lambda_U_Ctrl [V]	14.00	14.00
Lambda_Vbat_Mon [V]	13.98	
SC_Supply_Mon [V]	12.04	
Vbat_HC_FIU_Mon [V]	14.06	

Safety

Watchdog_State_Mon [0|1]	●	☑ ON/OFF

EPS Ignition Ctrl

EPS_Ignition_Ctrl [0|1]	●	☑ ON/OFF
EPS_Ignition_TH_Mon [V]	13.97	
EPS_Power_Ignition_Ctrl [0|1]	●	☑ ON/OFF
EPS_Power_Ignition_Mon [V]	14.02	

E-VGR Ignition Ctrl

VGR_Ignition_Ctrl [0|1]	●	☐ ON/OFF
VGR_Ignition_TH_Mon [V]	0.00	
VGR_Power_Ignition_Ctrl [0|1]	●	☐ ON/OFF
VGR_Power_Ignition_Mon [V]	3.84	

> “We rely on the dSPACE DS5203 FPGA board and the XSG Electric Components library for the electric motor simulation of our steering systems. The achieved performance and simulation quality are best suited for our applications.”
>
> *Tetsuya Nozawa, JTEKT*

serial interface) can read the ECU’s RAM so the testers are well-informed about the processes in the EPS ECU and can perform diagnostics tasks if necessary. Last but not least, the requirements managed in DOORS can be traced throughout the entire tool chain.

Conclusion and Outlook

The dSPACE simulator and the integrated load have proven their worth in the development project. The EPS electronic control units for steering systems of different customer vehicles were successfully validated. Because the dSPACE product portfolio covers the entire V-cycle, the HIL tests can be coordinated with the results of tools for rapid control prototyping (MicroAutoBox) and production code generation (TargetLink). The simulator already contains a further FPGA board and an Electronic Load Module DS5381 for the simulation of a future electric motor for an electronically controlled variable gear ratio steering (E-VGR). To emulate the mechanical load of the steering system, and therefore the mechanical load of the simulated electric motor, the ASM Vehicle Dynamics model will be used in the future. The test system is planned to not only validate ECUs but also to be used in the early development phases to test and optimize functions as early as possible. We cooperate closely with dSPACE to develop suitable solutions. In this context, the need for additional tools arises, mainly to manage data seamlessly and ensure compatibility across the entire tool chain. This could be solved with the data management tool SYNECT®. ■

Hirozumi Eki,
Tetsuya Nozawa,
JTEKT

Executive Summary

The Japanese steering system manufacturer JTEKT develops its electric power steering (EPS) systems in compliance with ISO 26262. To track requirements throughout all development stages, JTEKT relies on a seamless tool chain, in which the requirements management tool IBM DOORS, the dSPACE Simulator and the test automation tool dSPACE AutomationDesk play an essential role.
For a complete verification, JTEKT performs ECU tests with the hardware-in-the-loop (HIL) method at power level. For this purpose, the simulator must precisely emulate the electrical behavior of the electric motor. This objective is achieved with a fast FPGA (field programmable gate array) processing platform (DS5203) in combination with the Electronic Load Module (DS5381). They guarantee a highly dynamic emulation of high motor currents in real time. With the installed tool chain, JTEKT develops EPS systems in compliance with ISO 26262 and validates them successfully.

Hirozumi Eki
Hirozumi Eki is office manager of System Development Office 2 at JTEKT in Shinpukuji-cho, Okazaki, Japan.

Tetsuya Nozawa
Tetsuya Nozawa is development engineer at JTEKT in Shinpukuji-cho, Okazaki, Japan.

Euro NCAP tests with virtual test drives

Stay Safe on the roads

As customer expectations rise and Euro NCAP requirements get tougher, the cost of developing advanced driver assistance systems threatens to become unmanageable. dSPACE offers the solution: a well-coordinated tool chain for function development, virtual validation and hardware-in-the-loop simulation.

Euro NCAP: Five-Star Safety

The rigorous evaluation criteria used in the European New Car Assessment Programme (Euro NCAP) are presenting carmakers with new challenges. Euro NCAP tests new vehicle models – for example, by performing crash tests – and gives each one a safety rating of up to five stars. The ratings cover four areas: occupant protection for adults and children, pedestrian protection, and safety assist.

Decisive: Active Safety Systems

Active safety systems are becoming increasingly important for obtaining the highest rating of five stars. In concrete terms, lane departure warning (LDW) systems and autonomous emergency braking (AEB) for urban use (AEB City) and rural use (AEB Inter-Urban) will be included in Euro NCAP assessments from 2014 onwards. Starting in 2016, autonomous emergency braking will include the detection of vulnerable road users (VRUs) such as pedestrians (AEB/VRU/Pedestrian). The challenge here is to design safety systems that react as intended in safety-critical situations – such responses are called true positives – but do not overreact and produce false positives, for example, by initiating emergency braking unnecessarily. Only the true positives are currently important for Euro NCAP. dSPACE provides an extensive test environment (figure 2), based on Euro NCAP test protocols, for validating active safety systems by means of simulation.

> By performing Euro NCAP tests in the simulation, active safety systems can be evaluated in early development phases.

Testing in Accordance with Euro NCAP

ModelDesk includes a library of ready-to-use Euro NCAP test scenarios that comply with Euro NCAP test protocols for use cases such as AEB City, AEB Inter-Urban and AEB VRU/Pedestrian. Figure 1 shows selected test protocols. Note that the Euro NCAP definitions for active pedestrian protection scenarios have not yet been finalized (information valid as of April 2014). Figure 2 gives an overview of the tool environment for validating the necessary ECU software by MIL/SIL simulations. The associated AutomationDesk project and the visualization of a test scenario in MotionDesk are shown in figure 3. AutomationDesk has a catalog of preconfigured tests that have been designed for easy handling. After loading a test project, users can select the ECU functions to be tested (autonomous braking, collision warning), either together or individually, and specify the planned testing depth according to the Euro NCAP categories. The entire project can be started with just a few clicks.

Automatic Tests and Test Reports

During automatic test execution, the individual components of the test environment are remote-

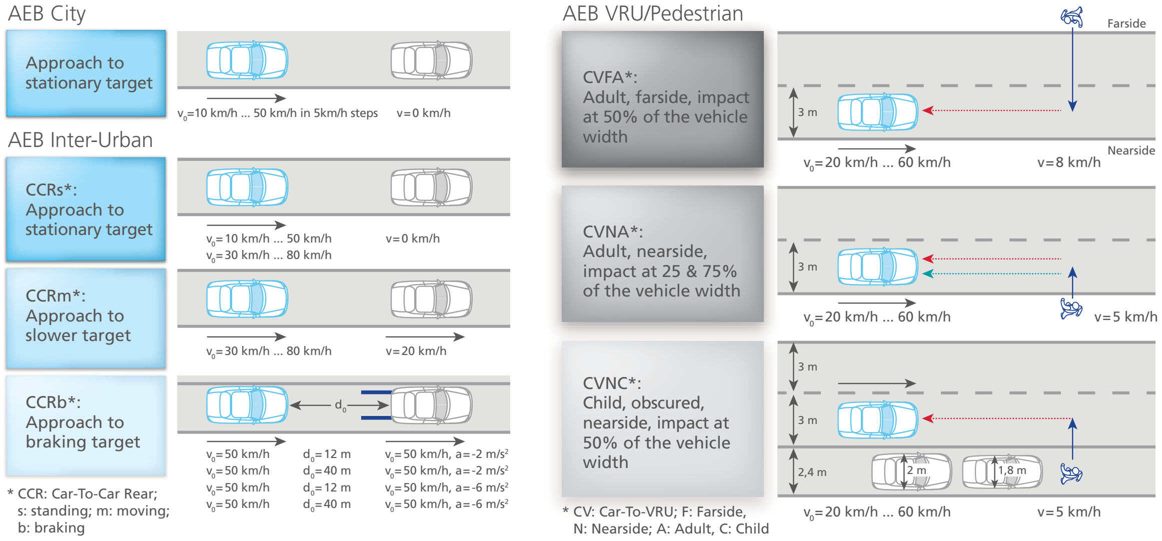
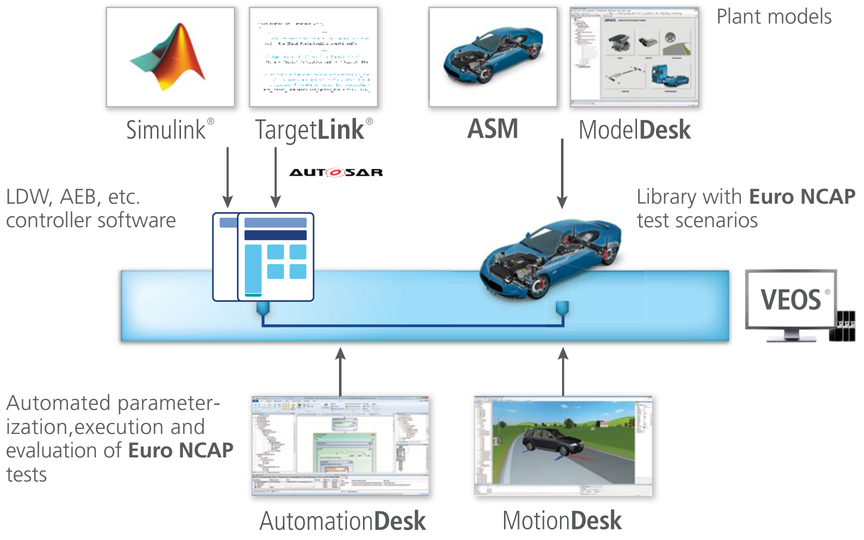


Figure 1: Euro NCAP test protocols for AEB City, AEB Inter-Urban and AEB VRU (vulnerable road user)/Pedestrian.

controlled via associated libraries. The process guarantees that each single test scenario is selected, parameterized, executed, evaluated, and precisely logged in compliance with Euro NCAP. MotionDesk shows developers the progress of each test so that they can assess its plausibility. When test execution completes, AutomationDesk generates a report comprising all the relevant information at three different levels of detail. First there is a concise overview (figure 4) of the total score for a test area (AEB Inter-Urban in this example). It includes graphical descriptions of associated test scenarios and a table of individual scores. Users can navigate to more detailed reports on individual test series from the result tree (on the left of the screenshot). These reports contain the main results of each test run in the test series together with the number of points awarded and the scores. There are also links to detailed reports on the individual test runs themselves. These contain all the details from the individual parameterization to

Figure 2: Simulation environment for executing virtual Euro NCAP tests. The associated AutomationDesk project and its visualization in MotionDesk are shown in figure 3.



AutomationDesk: Graphical descriptions of test sequences; custom libraries; automatic report generation

ASM (Automotive Simulation Models): Open MATLAB®/Simulink® models for simulating vehicles, road networks and the traffic environment

ModelDesk: Graphical user interface for defining test scenarios and parameterizing ASM

MotionDesk: 3-D animation software for the real-time visualization of traffic scenarios; object libraries for vehicles, roads, traffic signs, traffic lights, etc.

TargetLink: Automatic production code generation straight from Simulink®/Stateflow®

VEOS: Validation of ECU software and bus communication by stimulus tests and MIL/SIL/PIL simulation on a standard PC

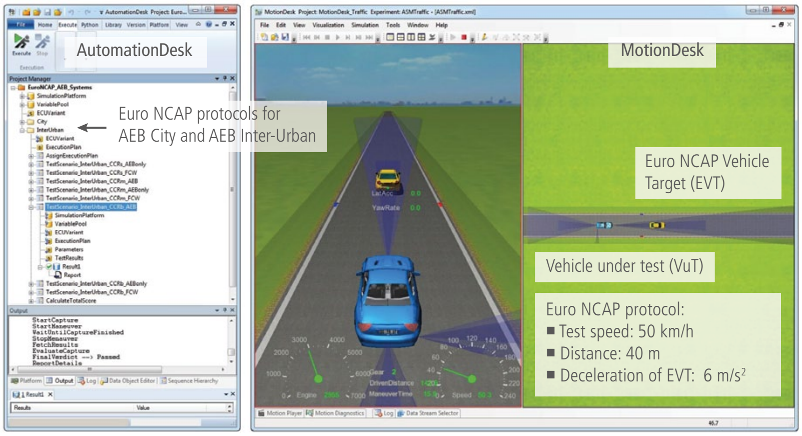


Figure 3: AutomationDesk project (left) and visualization in MotionDesk (right).

tables and graphics of the measurement results, as well as the final score.

Optimized Test Frame for Driver Assistance Systems

Test execution in AutomationDesk employs a test frame that was specially developed for validating driver assistance systems. The frame is extremely easy and convenient to use, and subsequent tests can also be based on it with very little additional effort. Once a user has defined the test scenarios in ModelDesk, it basically takes just three steps to create the final test.

The first step is to configure the test frame so that it fits the test environment, in other words to define the test platform, the signals to be measured, the test parameters, the final ModelDesk test scenario, and so on.

In the second step, the test scenario can be parameterized separately for each individual test run (e.g., speed of the vehicle under test).

For the third step, the test frame provides a special area where users can integrate their own test evaluation and logging setups.

All the other steps necessary for test execution – selecting and activating the test scenarios in ModelDesk, downloading test parameters to the platform, maneuver control, data capture, etc. – have already been performed and the results integrated into the test frame. They automatically start running in the background at the appropriate times.

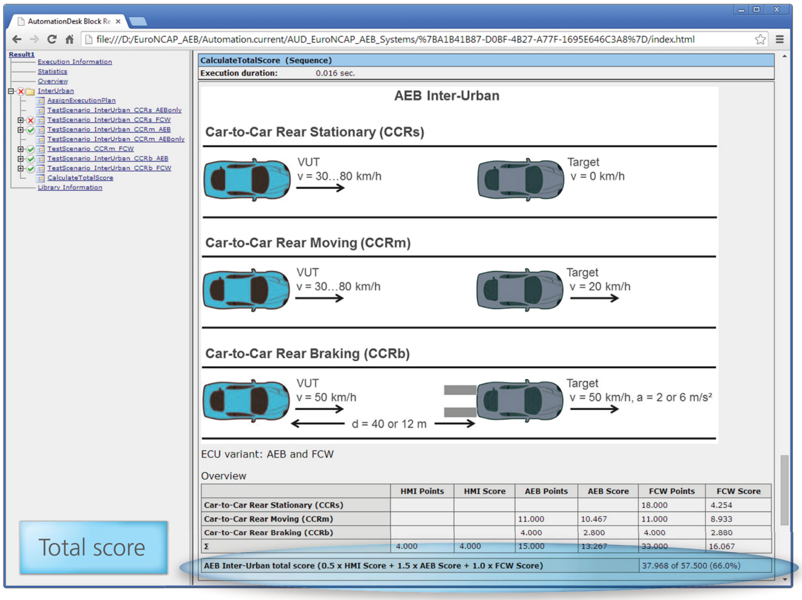
Thanks to this development environment, test developers can concentrate on their essential tasks and need no other special knowledge, such as how to handle tool automation.

Testing More Than Euro NCAP

System behavior can also be studied at its extreme limits and false positives can be detected by varying appropriate parameters in the available test scenarios. This especially applies to the design of systems for pedestrian detection, where aspects such as pedestrians' direction and speed of walking can also be varied in addition to the Euro NCAP specifications.

The Automotive Simulation Models (ASM) include sensor and object models for simulating these aspects for early system evaluation. Users can then determine the rate of false positives very early on and modify the software accordingly during system design. Yet another feature, planned for the upcoming version of MotionDesk, is very realistic animation of the movements of adults and children. This is particularly important for testing camera-in-the-loop HIL systems. ▪

Figure 4: Euro NCAP test report generated by AutomationDesk.



CalculateTotalScore (Sequence)

Execution duration: 0.016 sec.

AEB Inter-Urban

Car-to-Car Rear Stationary (CCRs)

VUT v = 30…80 km/h

Target v = 0 km/h

Car-to-Car Rear Moving (CCRm)

VUT v = 30…80 km/h

Target v = 20 km/h

Car-to-Car Rear Braking (CCRb)

VUT v = 50 km/h

Target v = 50 km/h, a = 2 or 6 m/s²

d = 40 or 12 m

ECU variant: AEB and FCW

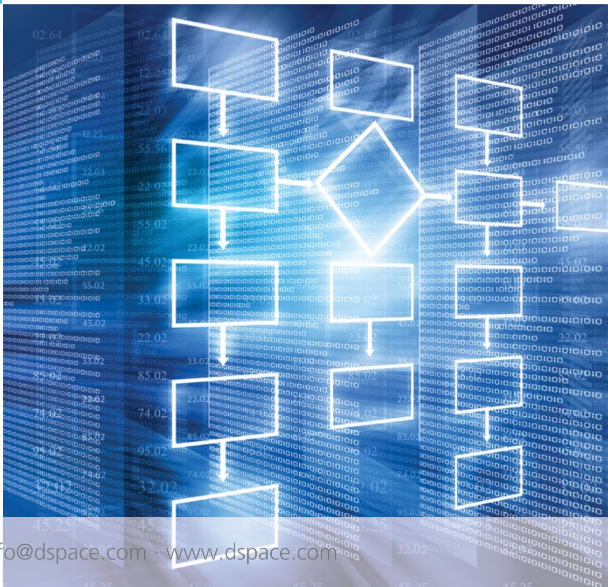
Overview

	HMI Points	HMI Score	AEB Points	AEB Score	FCW Points	FCW Score
Car-to-Car Rear Stationary (CCRs)					18.000	4.254
Car-to-Car Rear Moving (CCRm)			11.000	10.467	11.000	8.933
Car-to-Car Rear Braking (CCRb)			4.000	2.800	4.000	2.880
Σ	4.000	4.000	15.000	13.267	33.000	16.067
AEB Inter-Urban total score (0.5 x HMI Score + 1.5 x AEB Score + 1.0 x FCW Score)					37.968 of 57.500 (66.0%)	

Solving the Complexity Puzzle

More and more extensive functionalities are being developed in work groups that consist of a large number of software developers from various development partners – a great challenge for the development process. A reliable tool chain for efficient, model-based software development is therefore crucial. Simulink/TargetLink and the tools by Model Engineering Solutions provide a tailor-made solution.

How to beat complexity and build consistency – even in large-scale distributed development



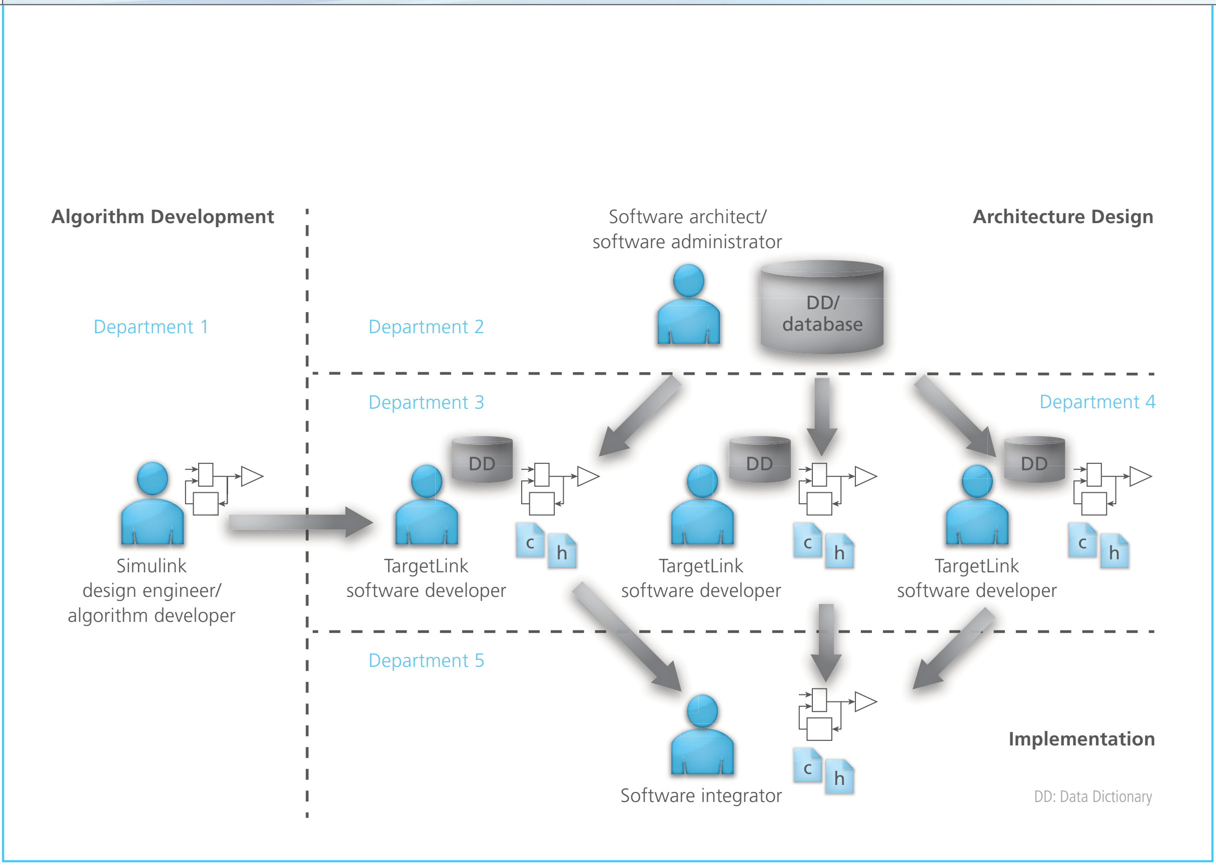


Figure 1: Distributed development in large teams. The model is defined in the domains Architecture Design and Algorithm Development and is further enriched by the development of subfunctions. The results are then aggregated, validated and implemented on an ECU.

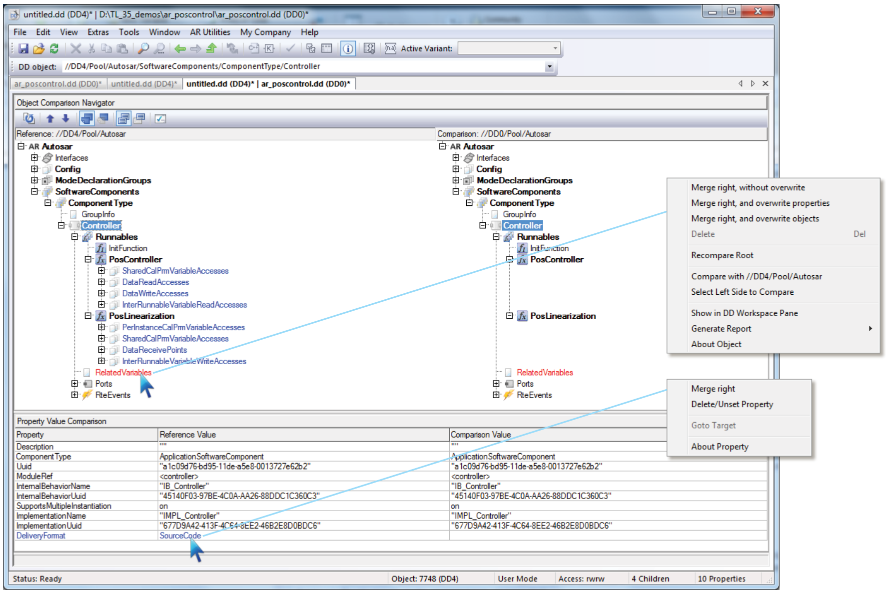
Challenge: Distributed Development

The Simulink®/TargetLink® models, the executable specifications of the software functions, generate not only code but also other artifacts, such as A2L files, AUTOSAR XML files and software documentation. If design and automatic production code generation apply only to individual software components and functions, developers do not detect inconsistencies until they integrate the components. Often, mechanisms for testing previous development steps do not exist. This problem is becoming more and more critical because vehicle functions are increasingly complex and require the development environment to be distributed across many work groups. To make modular, distributed development of extensive functionalities efficient, developers need to adapt development mechanisms and modify a tool chain tailored to Simulink/TargetLink.

Method	Benefit
Modeling guidelines	▪ Consistency ▪ Lower susceptibility to errors ▪ Less rework
Reuse (libraries, referenced models)	▪ Modular development ▪ Clarity due to model organization and hierarchy ▪ Reduced development effort by reusing the same models
Single source specifications	▪ Easier exchange between development team members due to software and interface specifications in the Data Dictionary
Incremental code generation	▪ Quicker reviews ▪ Faster code generation ▪ Easier software integration and testing
Code generation from the Data Dictionary	▪ Generation of shared variables in one file
Diff&Merge mechanisms via TargetLink Data Dictionary and Model Compare	▪ Traceability of changes to interface definitions and the model
Complexity analysis with M-XRAY	▪ Indication of appropriate model partitioning

Modeling Guidelines Improve Consistency and Mitigate Susceptibility to Errors

Simulink/Stateflow® provide many modeling possibilities, but not all of them can be used for efficient production code generation. Modeling guidelines lowering the risk of faulty models are especially important when many developers work on the same software. Adhering to these guidelines minimizes the amount of reworking needed, harmonizes modeling styles, simplifies testing and serves



Property Value Comparison

Property	Reference Value	Comparison Value
Description	""	""
ComponentType	ApplicationSoftwareComponent	ApplicationSoftwareComponent
Uuid	"a1c09d76-bd95-11de-a5e8-0013727e62b2"	"a1c09d76-bd95-11de-a5e8-0013727e62b2"
ModuleRef	<controller>	<controller>
InternalBehaviorName	"IB_Controller"	"IB_Controller"
InternalBehaviorUuid	"45140F03-97BE-4C0A-AA26-88DDC1C360C3"	"45140F03-97BE-4C0A-AA26-88DDC1C360C3"
SupportsMultipleInstantiation	on	on
ImplementationName	"IMPL_Controller"	"IMPL_Controller"
ImplementationUuid	"677D9A42-413F-4C64-8EE2-46B2E8D0BDC6"	"677D9A42-413F-4C64-8EE2-46B2E8D0BDC6"
DeliveryFormat	SourceCode	

Figure 2: Comparison of different software specifications.

as a reference for reviews. It is also easier for development teams to exchange models and functionalities. Tools for automated guideline checks, such as the MES Model Examiner®, check for guideline violations and correct them.

Partitioning and Reusing Models

The single source principle is an integral part of the distributed development process. 'Single source' means that the same model is used in different development phases, from design to closed-loop control to integration. Simulink/TargetLink realize these mechanisms by:

- Using Simulink library mechanisms to reuse multi-instantiable model parts or
- Using model referencing mechanisms to integrate models into other models

Simple Exchange and Administration

In large development teams, tasks such as function development, software architecture and administration, software development and integration are rarely carried out by only one person. Rather, a large number of team members access the same information (figure 1). Since design engineers predominantly exchange, edit, and save specifications, these have to be consistent. TargetLink offers a specialized tool, the TargetLink Data Dictionary (TL-DD), which by default supports various exchange formats, such as XML or AUTOSAR XML. The data objects in the model and in the Data Dictionary are linked, so that the algorithm is separated from the data, and the

data in the model and in the Data Dictionary are synchronous.

Powerful Incremental Code Generation

Incremental code generation is another core method of distributed, model-based development. Code is generated incrementally for the individual software units. The repercussions that changes to a small function have on the overall software functions are kept to a minimum because the software units are independent from each other. Code only has to be generated for the unit that has been modified, while the rest remains unchanged.
Manual reviews are therefore less time-consuming and code generation time is held to a minimum. This makes it possible to develop large functionalities more efficiently and faster.

Generating Code from the Data Dictionary

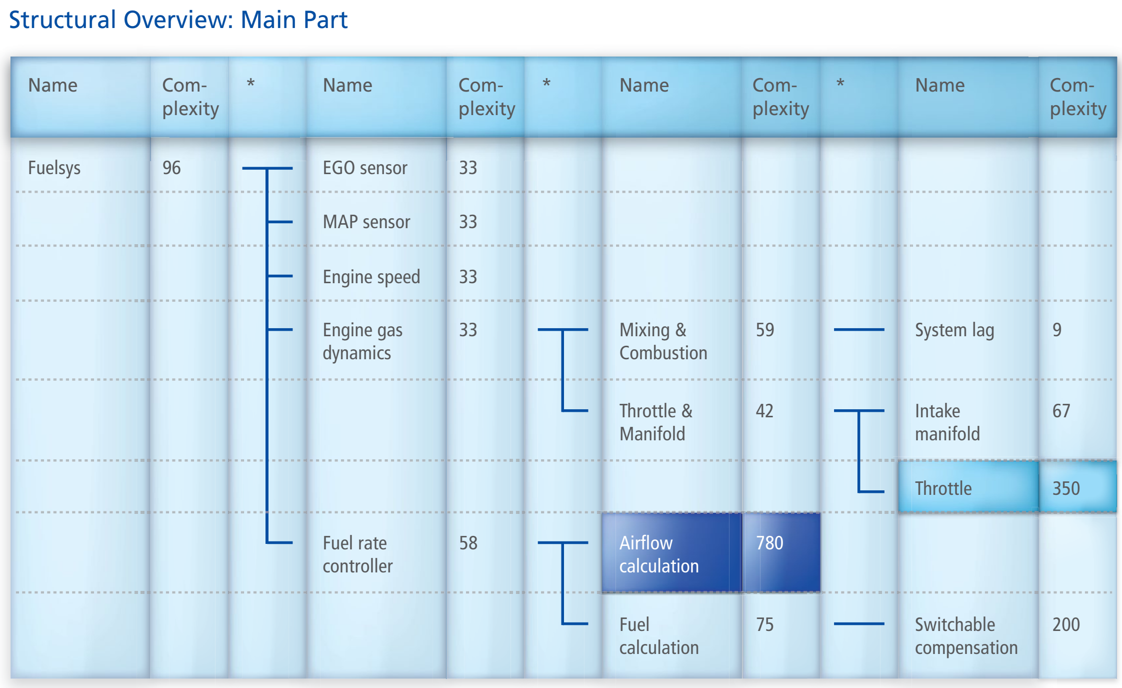
Code is generated directly from the Data Dictionary, independently of the model, and global or shared variables from the Data Dictionary are generated to a file. This method is used when:

- A file is created that contains all global variables, such as interface variables, and specifies the access rights to them.
- All calibration parameters of different functions are generated in a single calibration parameter file
- Variables that are used in automatically or manually generated legacy code are generated in a file.

Efficient Diff&Merge Mechanisms

When developing new software, design engineers have to be able to identify changes, especially when different departments and suppliers are involved. Design engineers and integrators exchange the modified software artifacts.
A reliable tool chain that identifies modifications is therefore indispensable. TargetLink Data Dictionary has the mechanisms needed to compare different versions and display changes (figure 2). These modifications can then be traced back to the model to see their effects on it. dSPACE's Model Compare, for instance, provides convenient and comprehensive functionalities to compare models.
DD mechanisms update interface definitions automatically to ensure

Figure 3: Analysis report of M-XRAY.



Structural Overview: Main Part

Name	Complexity	*	Name	Complexity	*	Name	Complexity	*	Name	Complexity
Fuelsys	96		EGO sensor	33						
			MAP sensor	33						
			Engine speed	33						
			Engine gas dynamics	33		Mixing & Combustion	59		System lag	9
						Throttle & Manifold	42		Intake manifold	67
									Throttle	350
			Fuel rate controller	58		Airflow calculation	780			
						Fuel calculation	75		Switchable compensation	200

consistency when changes are made (figure 2).

Validating Model Architecture

When TargetLink models are used for distributed development, they have to be divided into subfunctions and subsystems. The complexity of the individual subsystems must be kept to a minimum to reduce the number of possible errors and to ensure subsystems are readable and maintainable. At the same time, this approach fulfills the requirements of safety standards, such as ISO 26262, which call for low complexity. Model complexity can be checked automatically by the MES Model Examiner AddOn M-XRAY. M-XRAY calculates and evaluates the complexity of the overall model and the individual subsystems. In addition, it collates all the model metrics relevant for a qualitative evaluation of TargetLink models (see Excursion on model metrics). M-XRAY generates an analysis report (figure 3) that gives an overview of the model hierarchy and the complexity of each subsystem. This makes it easy to evaluate a model's complexity and to identify particularly complex subsystems.

Reference values for local complexity

Reference value	Evaluation
MV ≥ 750	High
MV < 750	Medium
MV < 300	Low

Summary

There are many efficient methods to control complexity and consistency in distributed development which can also be used by large development teams. Model partitioning, incremental code generation, and tools supporting change tracking are the ingredients for success. Design engineers can use measurements and metrics to evaluate the partitioning. With this approach, design engineers can develop extensive functionalities more easily and exchange the developed subfunctions more efficiently and with less errors.

Excursion

Model Metrics

Using metrics, developers can compare TargetLink models and evaluate their complexity and quality. Safety standards, such as ISO 26262, stipulate that the complexity of safety-critical models must be evaluated (see ISO 26262-6, §5.4.7, table 1). Model metrics can also be used to estimate the effort required for testing and reviews. By capturing metrics values for different development stages, developers can also monitor a model's development and identify particularly complex and error-prone model parts very early on.

Metrics such as the number of blocks, *modeling depth*, interface width, or cyclomatic complexity are also used to measure model complexity. However, these metrics are based on programming concepts and are not often suited for evaluating models. An evaluation of the cyclomatic complexity of models, for example, is not very informative due to the data flow orientation in Simulink.

The measurement of *model volume* (MV), derived from Halstead complexity measures, is establishing itself in the industry as an important way to evaluate model complexity. This measurement allows developers to evaluate model complexity because it includes not only model blocks but also the links between blocks, their weights and their own complexities, and the signals used to link blocks.

The MES Model Examiner® and the M-XRAY AddOn can be used to analyze and evaluate TargetLink models with model metrics. M-XRAY analyzes the models and calculates their volume and all relevant metrics values. It then presents the results in a compact, structured table. This tool therefore makes it possible to efficiently calculate complexity distribution in a model and keep it to a minimum.

Literature:
Stürmer, I., Pohlheim, H., Rogier, T.: "Calculation and Visualization of Model Complexity in Model-based Design of Safety-related Software", (in German) in Keller, B. et. al., Automotive - Safety & Security, Shaker, pp. 69-82, 2010.

SYNECT, the new dSPACE data management solution with a focus on model-based development, has been on the market since October 2012. It started with one module for test management and one for integrated variant management. Now modules for managing signals, parameters and models are also available.

Mastering the Data Wave

The interest in SYNECT is enormous. This is because model-based development is being used everywhere, every day to develop new functions. Networked functions that need to be adapted to fit hundreds of vehicle variants, with thousands of parameters and signals and very many software modules. All of this happens in development processes where there are multiple stages and many developers and testers participating. Only clean processes and clean data management can guarantee that work is done systematically and reliably. This is where SYNECT comes in. SYNECT speaks the language of the user. Its modules for test management, signal and parameter management, and model management provide direct support for everyday development tasks. For example, engineering tools can be connected directly to SYNECT, and data exchange via key standards and common file formats is supported.

Modular and Scalable

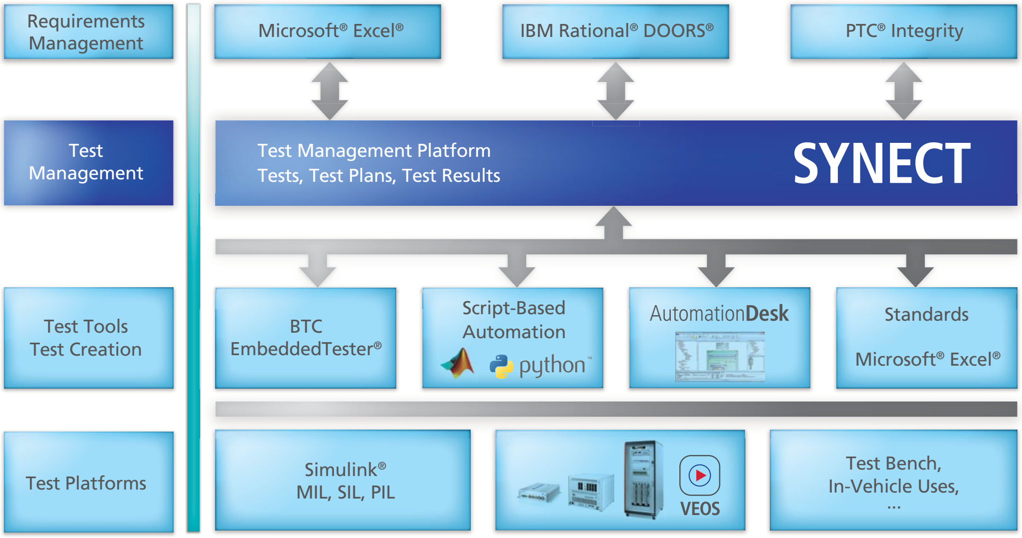
SYNECT is modular in design and you can add extensions one after the other to build up a comprehensive central data management system. To handle the many variants that accumulate during development, SYNECT provides systematic support. For example, you can explicitly specify variant dependencies according to a configurable model variant and then evaluate and consider these dependencies when the data is used later.
SYNECT acts as the data management center and provides fine-grained versioning as well as comprehensive user and rights management so that teams can reuse data and work together.

Cross-Phase Test Management

SYNECT Test Management provides comprehensive support for testing functions, software, and electronic control devices throughout the various phases of the development process. It focuses on supporting MIL, SIL, PIL and HIL tests, and also manual tests. One important aspect is that SYNECT can be integrated seamlessly into existing processes and tool environments. It not only supports test automation tools such as dSPACE AutomationDesk, but also standard exchange formats such as Microsoft® Excel® and XML. Customer-specific test tools, test formats and script-based solutions can be connected via a generic interface. Test cases can be specified directly in SYNECT or imported from existing test specifications and test implementations. This means that the start-up phase for learning how to work with SYNECT Test Management is quick and fuss-free. SYNECT not only lets you manage test cases

Impressions from SYNECT users:

> "The seamless integration of SYNECT and AutomationDesk was a decisive advantage when we were extending our test automation environment. The process for compiling test execution plans and evaluating test results runs efficiently and intuitively."
>
> *Alessandro Recca, ABB Switzerland*



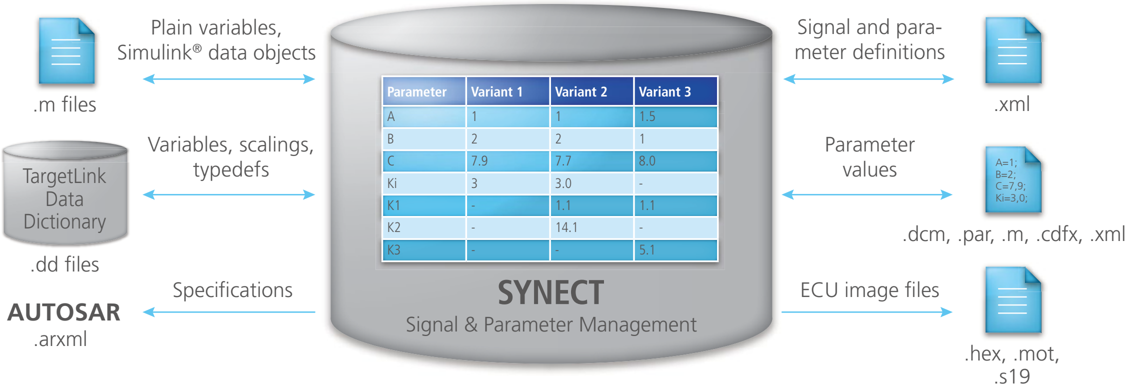
Because it supports different test tools and platforms, SYNECT Test Management can be used flexibly in all phases of the development process.

clearly, but also gives you the ability to plan test case executions from a central location and initiate them directly. After the tests are executed successfully, the test results are fed back into SYNECT and stored for traceability throughout all project phases. The interface to requirements management is especially important. Requirements can be linked to the test cases to support requirements-based test workflows and to ensure traceability from the requirements to the test cases and test results. The direct overview in SYNECT shows what test coverage level of the requirements was achieved.

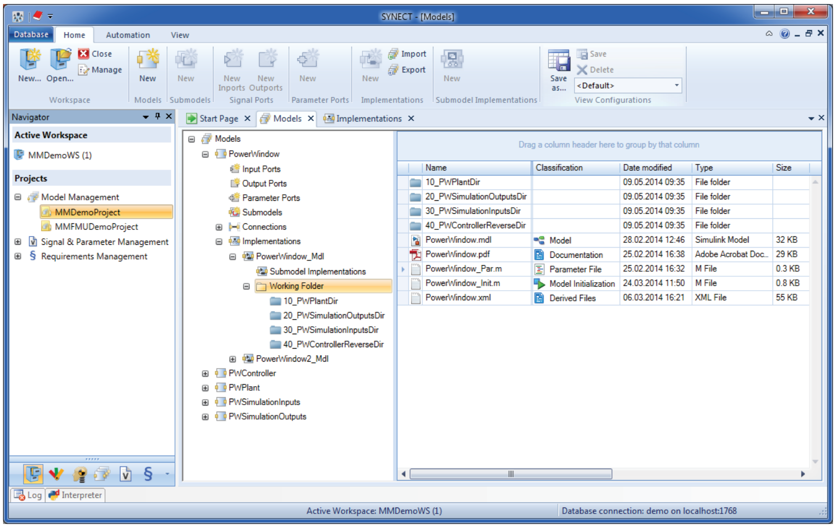
Managing Model Variety Efficiently

The many function models and simulation models that are created in embedded electronics development can be managed centrally and reused easily by different teams and in different projects. SYNECT handles models not just as black boxes by using files, but uses metadata and a fine-grained level of detail that includes interface signals, parameters and submodels. This model structure information is generated automatically via model analysis when the new model is imported into SYNECT. As a result, the information is more transparent because the model management functionality already shows you the model's interfaces and parameters. You do not have to switch to the modeling environment. Integrating models

Numerous import and export interfaces support integrated signal and parameter management and use throughout the entire development process.



Parameter	Variant 1	Variant 2	Variant 3
A	1	1	1.5
B	2	2	1
C	7.9	7.7	8.0
Ki	3	3.0	-
K1	-	1.1	1.1
K2	-	14.1	-
K3	-	-	5.1



<Default>

Drag a column header here to group by that column

Name	Classification	Date modified	Type	Size
10_PWPlantDir		09.05.2014 09:35	File folder	
20_PWSimulationOutputsDir		09.05.2014 09:35	File folder	
30_PWSimulationInputsDir		09.05.2014 09:35	File folder	
40_PWControllerReverseDir		09.05.2014 09:35	File folder	
PowerWindow.mdl	Model	28.02.2014 12:46	Simulink Model	32 KB
PowerWindow.pdf	Documentation	25.02.2014 16:38	Adobe Acrobat Doc...	29 KB
PowerWindow_Par.m	Parameter File	25.02.2014 16:32	M File	0.3 KB
PowerWindow_Init.m	Model Initialization	24.03.2014 11:50	M File	0.8 KB
PowerWindow.xml	Derived Files	06.03.2014 16:21	XML File	55 KB

Active Workspace: MMDemoWS (1)

Database connection: demo on localhost:1768

SYNECT manages models centrally and traceably with metadata, interface signals, parameters and associated files.

later on can be simplified because the interfaces can be defined at a central location, independently of the actual implementation. There is also a separate module for seamless signal and parameter management throughout the entire development process, independent of model management. Of course, SYNECT also manages the actual model files and all related files such as initialization scripts and model documentation. The hierarchical representation of the models not only makes it possible to organize submodel reuse but is also the basis for creating configurable integration models in SYNECT. The interface information is helpful for checking whether models can be used with each other. This check often relies not just on interface information but also on additional context- and scenario-specific information – such as which variants or scenarios a particular model is suitable for.

Adding Metadata

The flexible metadata management provided by SYNECT lets you expand the information that is stored in the individual models. This metadata not only makes it easier to search for specific models, but can also be evaluated in integration scenarios afterwards. SYNECT's model management functionality currently particularly supports MATLAB®/Simulink® models and functional mock-up units (FMUs) in accordance with the Functional Mock-up Interface (FMI) standard. dSPACE will continue to expand its range of supported model formats. ■

Video:
dSPACE SYNECT
http://www.youtube.com/watch?v=ZebVW9kwjOI

Impressions from SYNECT users:

> "SYNECT is user-friendly and meets the special requirements that test managers have. Because of its interfaces to DOORS® and AutomationDesk, SYNECT is the only central tool we need to plan, manage, and control all the test activities, and the automated tests in particular. Using the SYNECT Test Management module helps us track the test activities throughout all the various test levels and test benches."

Christian Trösch and Alexander Wiener,
Division Powertrain, Continental Corporation

Success in Series

From the very beginning, dSPACE has actively supported the development and widespread use of the AUTOSAR standard and has inspired many customers to work with dSPACE's AUTOSAR-supporting tools. The many successful projects and innovative developments encourage dSPACE to further boost AUTOSAR support. Joachim Stroop, Lead Product Manager and AUTOSAR expert at dSPACE, gives insight into the current activities.

AUTOSAR's success story and how the dSPACE tool chain supports the AUTOSAR standard

How has the AUTOSAR standard evolved from a dSPACE point of view?
After more than ten years of hard work, the standard is now a core component in many countries. AUTOSAR is a technologically mature and extensive standard that can be used internationally. But the standard has not been put into practice equally in all regions. Although the initial interest is still strong in Europe, the same cannot be said for other markets. But dSPACE remains internationally active and continues to create commonly accepted, state-of-the-art means for the development and exchange of software in vehicle development.

> Our customers have successfully used our products in production projects for a long time.

How important is the AUTOSAR standard for dSPACE?
The AUTOSAR standard has entered all traditional vehicle domains, from vehicle electronics, to vehicle dynamics, to the drivetrain. The standard also increasingly replaces established description formats, which now have a common basis. This is especially true for descriptions of bus system communication. Here, AUTOSAR is used more frequently than DBC, LDF, and FIBEX. This makes it possible to introduce new technology such as Ethernet into the vehicle. The dSPACE tool chain supports AUTOSAR standard developments in all stages.

How does dSPACE contribute to AUTOSAR organizational work?
From day one, in 2004, dSPACE has been a premium member of the AUTOSAR partnership and has always been active in the AUTOSAR working groups. As a provider of tools for the development and testing of electronic control units, we use our know-how to promote standardization. We develop description formats for function components and software architectures, and work on communication protocols. We also support the efforts to use the well-established methods of rapid control prototyping and bypassing in AUTOSAR development projects.

How does dSPACE include the standard in its solutions?
We develop new applications that generate added value for our customers, all based on the AUTOSAR standard. dSPACE SystemDesk®, for example, makes developing software architectures and generating virtual electronic control units possible. These can be validated in early development phases using dSPACE simulation platforms such as VEOS®. This is only possible with the AUTOSAR standard. But the standard is also becoming a part of other tools to provide new possibilities created by the standard and to have a product range that can be used in all development stages. Some examples are AUTOSAR-compliant code generation with dSPACE TargetLink® and the AUTOSAR-compliant use of bus formats for configuring hardware-in-the-loop systems. Another example is rapid control prototyping in combi-

> AUTOSAR has entered all areas of traditional vehicle electronics.

> We want to create added value by adapting the development methods for real AUTOSAR ECUs to the virtual world.

nation with AUTOSAR modules and bypassing of AUTOSAR electronic control units. dSPACE invests not only in these tools but also in optimizing how our tools are put to use. Often, a small investment yields big returns. The experienced AUTOSAR experts at dSPACE Engineering are there to support these services.

Where can dSPACE products be used for AUTOSAR-compliant development?
The AUTOSAR standard covers many areas, and one provider alone cannot cover all of them. At dSPACE, the AUTOSAR standard focuses on the company's core competencies. Thanks to the advanced maturity of the AUTOSAR standard, dSPACE tools can be used together with the solutions of other providers in customer projects.

How successful is dSPACE with AUTOSAR?
For a long time, our customers have successfully used our hardware and software in production projects. The PSA article in this issue describes a striking example. PSA's approach has become common practice in Europe and is gaining importance across the globe. Virtual validation is a prominent and often-discussed topic in the automotive markets. dSPACE has recognized the potential for this application field very early and is considered to spearhead innovation in applications for front-loading tests – while ensuring a seamless transition to hardware-in-the-loop tests.

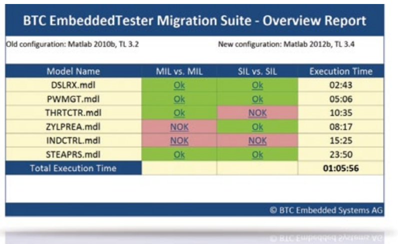
What does the future hold for AUTOSAR at dSPACE?
We are currently preparing Release R4.2 of the AUTOSAR standard, which our products will support as soon as possible. The quick support of new releases is a top priority for us. At the same time, we want to increase the added value for our customers, by adapting development methods for real electronic control units to the virtual world, for example. One solution is to introduce bypassing to virtual ECUs.

Thank you very much for your time, Mr. Stroop.

Joachim Stroop is Lead Product Manager for SystemDesk at dSPACE GmbH.

Fully Automatic Validation of Simulink/ TargetLink Version Changes

Changing to a new tool version regularly poses new challenges for users in model-based development. New product versions or modified IT infrastructures invariably mean that existing models and functions have to be migrated to a new version of MATLAB®, TargetLink® or Windows®. One important aspect of such updates is ensuring that they do not affect the simulation behavior of the model and production code. BTC Embedded Systems AG, a TargetLink Strategic Partner, provides an easy, highly integrated and completely automated solution for this: the BTC EmbeddedTester Migration Suite. The suite provides back-to-back testing with automatically generated test cases, based on TargetLink code, that guarantee complete test coverage. Back-to-back testing is not only an established, state-of-the-art method in model-based development, where (since the advent of the ISO 26262 standard) it is typically used to compare the model and the code – it is also the ideal solution for validating a tool migration. In the BTC EmbeddedTester Migration Suite, all the user has to do is specify the tool versions and enter a list of TargetLink models. When this has been done, all the other steps, such as test case generation and test case execution in both tool versions, are performed completely automatically. The suite finishes by producing a global HTML report clearly documenting the results of all the individual migration steps for all the models. ■



BTC EmbeddedTester Migration Suite - Overview Report

Old configuration: Matlab 2010b, TL 3.2

New configuration: Matlab 2012b, TL 3.4

Model Name	MIL vs. MIL	SIL vs. SIL	Execution Time
DSLRX.mdl	Ok	Ok	02:43
PWMGT.mdl	Ok	Ok	05:06
THRTCTR.mdl	Ok	NOK	10:35
ZYLPREA.mdl	NOK	Ok	08:17
INDCTRL.mdl	NOK	NOK	15:25
STEAPRS.mdl	Ok	Ok	23:50
Total Execution Time			**01:05:56**

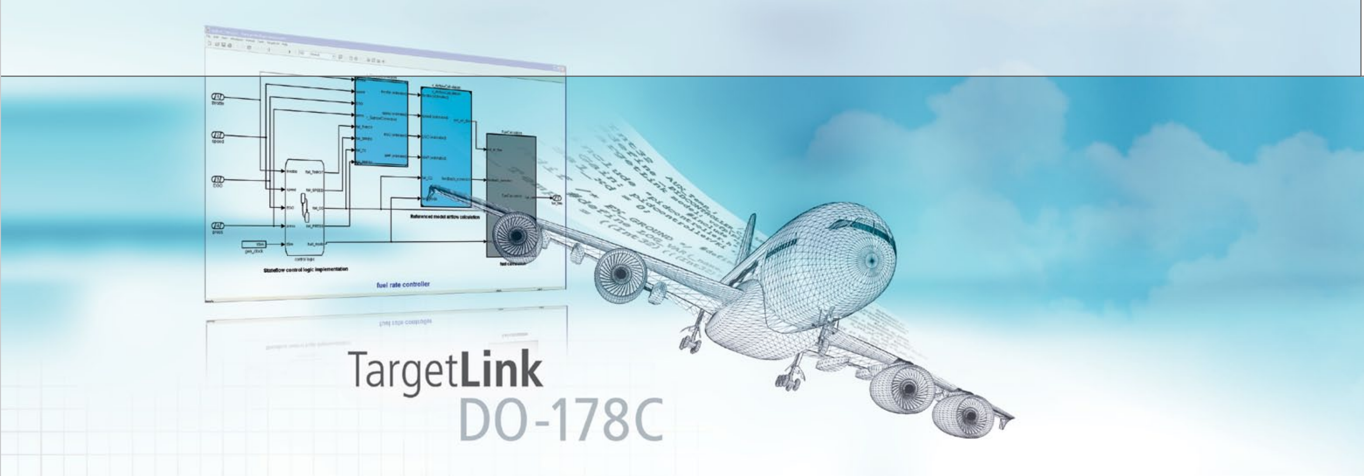
© BTC Embedded Systems AG

Report produced by the BTC EmbeddedTester Migration Suite.

SCALEXIO: Flexible System Extensions

dSPACE Release 2014-A includes two new extension options for the hardware-in-the-loop (HIL) system, SCALEXIO®. The first is multi-processor-unit (multi-PU) support, which allows two or more SCALEXIO systems to be connected to increase the available performance. One clearly organized ConfigurationDesk® project is used to distribute the individual simulation models to the available processors and also to configure the overall simulation model.

The second option is to couple the SCALEXIO Processing Unit with a DS1006/DS1005-based HIL system. Users can then continue working with existing systems for as long as possible to protect their investments, while at the same time benefiting from the innovations offered by the new SCALEXIO technology. Data is exchanged through a common interface that is addressed via Real-Time Interface blocks on the DS1006/DS1005-based side and via ConfigurationDesk on the SCALEXIO side. SCALEXIO acts as the master and synchronizes the overall system. ■



TargetLink for Aerospace: DO-178C Workflow Document Now Available

With the DO-178C as the future relevant standard for the development of software in aviation, model-based design and automatic code generation will have a solid base for use in the aerospace sector. The document DO-331, Model-Based Development and Verification Supplement to DO-178C and DO-278A, which is also part of the standard, was written specifically for this. dSPACE takes this into account by providing a workflow document that explains how to use TargetLink in a model-based tool chain for DO-178C-compliant projects. The workflow document describes how to meet the individual requirements or "objectives" of DO-178C/DO-331. It focuses not just on TargetLink itself but also on a complete model-based tool chain that can contain further third-party tools, for example, from TargetLink cooperation partners such as BTC Embedded Systems, AbsInt and Model Engineering Solutions. The workflow document is thus an important contribution towards simplifying the certification of TargetLink-generated code in DO-178C-compliant applications, addressing all criticality levels up to Level A.
You can obtain the approximately 60-page document from dSPACE Support. ■

Innovations in Modular Hardware

The New DS1007 PPC Processor Board

With its dual-core PowerPC architecture, the new DS1007 PPC Processor Board lets you calculate complex Simulink® models with low-latency I/O access for high closed-loop sampling rates. The board also features Gigabit Ethernet interfaces for host connection and for exchanging data between the real-time model and Ethernet-based devices.

In addition, a USB interface permits long-term data recording on a USB stick or USB hard drive. Complex Simulink models can be easily distributed onto the two processor cores by using the graphical user interface Real-Time Interface for Multiprocessor Systems.

AutoBox Soon for 48 V Electrical Systems Too

dSPACE AutoBox will soon feature an improved power supply (planned for late 2014 to early 2015). At 300 W it will have twice as much performance as its previous model, greater efficiency, and an improved startup behavior. The new AutoBox will also support 48 V systems in addition to the 12 V and 24 V systems already supported by the previous model. ■

ADAS Blocksets Can Now Be Used with VEOS

dSPACE provides special Simulink® blocksets for in-vehicle function development and hardware-in-the-loop testing of advanced driver assistance systems. These include the **ADASIS v2 Horizon Reconstructor Blockset** and **ADAS RP Blockset** for accessing electronic horizon data and the **ADTF Blockset** for connecting to EB Assist ADTF. The blocksets now also support PC-based simulation with VEOS® in addition to dSPACE real-time platforms. This gives end users the opportunity to use open-loop and closed-loop simulation to validate driver assistance functions on standard PCs at an early stage. They can also use the same function models, plant models, and operating, visualization, and testing tools seamlessly throughout the development process. Simulating driver assistance systems on PCs especially provides a high degree of flexibility for running through different maneuvers and test scenarios in a short time. The result is that the function models already have a high level of maturity in early development stages, which saves development time and costs. ■

Real-Time Simulation of Electrical Circuits

The new simulation model library **Power RealTime Library** boosts the performance of the real-time calculation of electrical models developed in SimPowerSystems (MATLAB®/Simulink®). The Power RealTime Library not only offers a powerful real-time simulation of SimPowerSystems models on the dSPACE platforms, it is also a tool that conveniently handles real-time code generation and provides interfaces optimized for dSPACE I/O boards. The library contains mean-value models for power electronics bridge circuits used for motor drive applications. These models precisely emulate the switching of power semiconductor devices such as IGBT, MOSFET etc. Simulation speed is increased by pre-calculation of model states and optimized model distribution across several computational nodes. Library functions for easy integration of models in asynchronous rasters avoid interference effects. The library can be used to develop automotive systems such as mild or full hybrid drives, electric brake systems, drivetrain actuators, electric power steering (EPS), auxiliary units (oil and water pump), etc. The library can also be used for the development of energy, motor drives, and automation technology. ■

Virtual Traffic

ASM Traffic, the simulation model for traffic and traffic environment simulations, has specifically been enhanced to support the development and test of modern advanced driver assistance systems (ADAS). The new version, **ASM Traffic 3.0**, one of the Automotive Simulation Models (ASM), has all the features necessary for using simulation to test active safety systems, such as emergency brake assistants, and to demonstrate their capabilities early in the development process. The model simulates a test vehicle, urban and rural road networks, a large variety of fellow vehicles, and traffic environment objects such as pedestrians and traffic signs. Virtual sensors scan the simulated environment and send signals to the driver assistance systems. With ASM Traffic 3.0 you can simulate collision-prone traffic scenarios repeatedly to test the functionality of active safety systems and driver assistance systems.

Complex Traffic Scenarios

ASM Traffic provides convenient user interfaces that help you define road networks, traffic signs, traffic vehicles, and the sensors of the test vehicle. This opens up flexible ways to act out complex traffic scenarios on a PC or a hardware-in-the-loop (HIL) simulator. Sample videos on the dSPACE ASM video channel clearly demonstrate the workflows of ASM Traffic and what ASM Traffic is capable of doing:

- Intersection Traffic – Shows a test scenario for validating intersection assistants
- Autonomous Parking – Demonstrates a parking assistance system

Application examples as a video: dSPACE ASM Video Channel

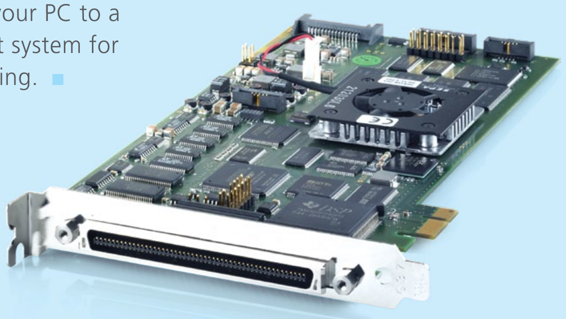
MicroAutoBox Embedded PC with an Intel Core i7

The Embedded PC for MicroAutoBox II is now available as a new, more powerful variant. Equipped with an Intel® Core™ i7 processor (dual-core, 2 x 1.7/2.8 GHz), 8 GB DDR3 RAM and 128 GB of flash memory (64 GB integrated, 64 GB flexible via CFast card), the Embedded PC provides the processing power you need for especially demanding applications. Ideal areas of application include systems for driver assistance, infotainment, telematics and image processing. It has several interfaces and also a temperature-controlled fan. You can choose between using a Windows® 7 (already included) and an Ubuntu Linux operating system. When combined, the MicroAutoBox II for real-time prototyping and the Embedded PC form a compact, complete system made of two powerful hardware units. The earlier fanless Embedded PC variant with an Intel® Atom™ N270 processor is still available. ■

DS1104 R&D Controller Board now with PCIe (PCI Express) host interface

The DS1104 R&D Controller Board is based on Power PC technology and its set of I/O interfaces makes it an ideal solution for developing controllers in various fields, such as drives, robotics, and aerospace. The board can be installed in virtually any PC with a free PCI or PCIe slot and is used in many university laboratories.

Now equipped with a PCIe host interface it upgrades your PC to a powerful development system for rapid control prototyping. ■

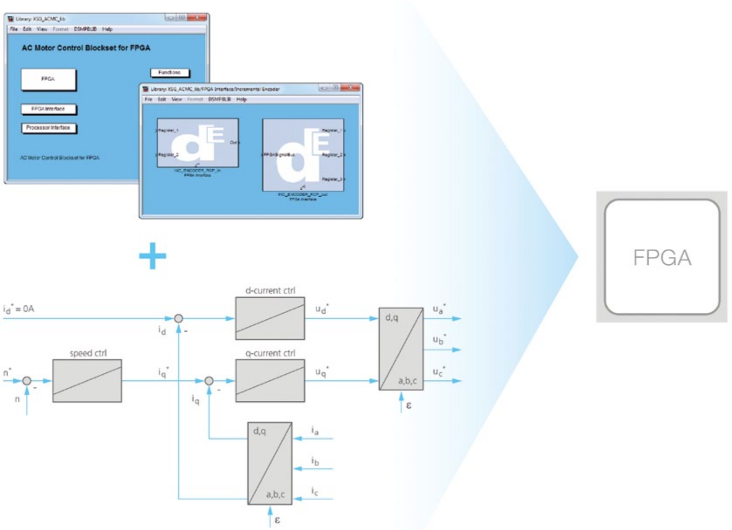


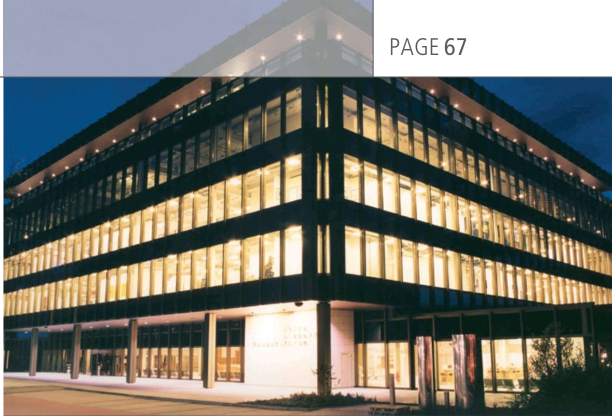
Quick and Easy: Programming Electric Drives Directly on FPGA

The new **XSG AC Motor Control** (XSG = Xilinx® System Generator) Library software is used for developing control systems for electric drives. The library enables engineers to program Field Programmable Gate Arrays (FPGA) for use in control tasks with dSPACE MicroAutoBox® II and SCALEXIO® systems, as well as modular dSPACE hardware. By moving control loops onto reconfigurable hardware, engineers for the first time ever can achieve execution speed of over 100 kHz and conveniently develop highly dynamic electric drive systems for applications demanding precise position and speed. The XSG AC Motor Control Library addresses these requirements with powerful capabilities for high-performance evaluation of various types of position sensors, including encoders, resolvers etc., and flexible generation of Pulse Width Modulation (PWM) signals. Developers can flexibly combine library components using the dSPACE RTI FPGA Programming Blockset software.

At a glance:

- FPGA library for fast electric drive controls
- Powerful position sensor evaluation
- Flexible PWM generation ■





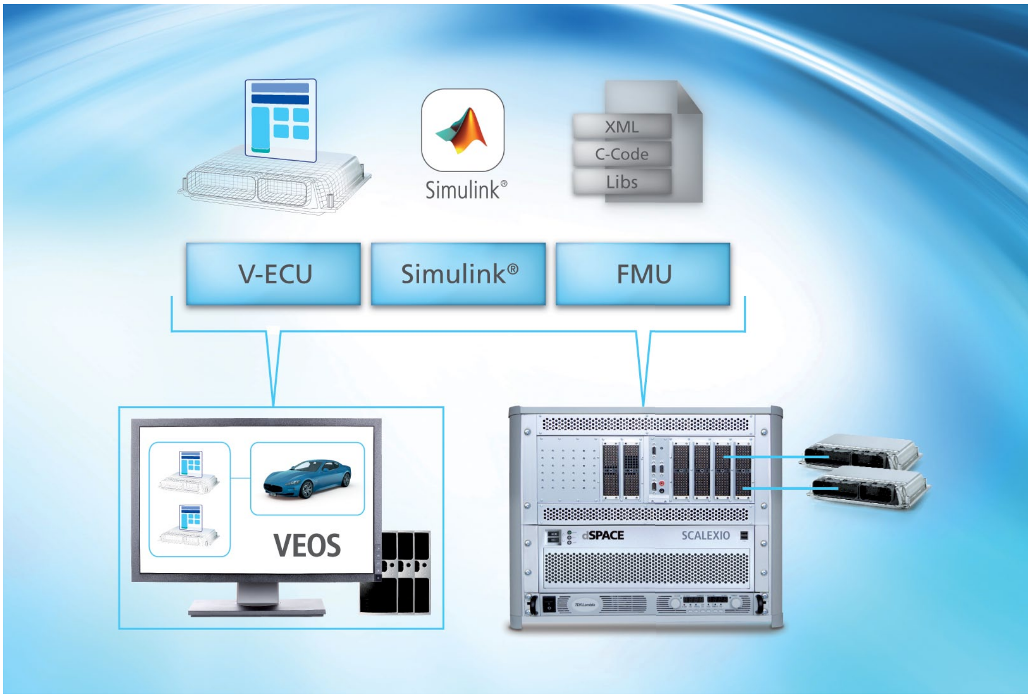
2014 IEEE Vehicular Networking Conference Paderborn

Vehicular networking and communication systems are an area of great importance in our increasingly connected and mobile world. Effective vehicular connectivity techniques can significantly enhance the efficiency of travel, reduce traffic incidents and improve safety, mitigate the effects of congestion, and overall provide a more comfortable traffic experience. As a step towards this goal, the 2014 IEEE Vehicular Networking Conference seeks to bring together researchers, professionals, and practitioners to present and discuss recent developments and challenges in vehicular networking technologies and applications.
This year's conference will be held at the Heinz Nixdorf MuseumsForum, the world's largest computer museum, in Paderborn, Germany, from December 3-5. It is organized by the dSPACE-endowed Professorship for Distributed Embedded Systems at the University of Paderborn. For details on submitting papers and other information, visit:
www.ieee-vnc.org ■

FMI Support for VEOS, SCALEXIO, and SYNECT

dSPACE's PC-based simulation platform VEOS® and hardware-in-the-loop (HIL) test system SCALEXIO® now also support the Functional Mock-up Interface (FMI) standard. This standard ensures that models from different suppliers and with different modeling approaches (physical or signal-based, for example) integrate smoothly into existing simulation models and can be shared with other users. This gives users the flexibility to choose which model is best suited to a particular project. Furthermore, SYNECT® Model Management now supports model implementations based on FMI.

For further information on FMI, visit *www.dspace.com/go/fmi* ■



Please let us know your opinion on the quality of the dSPACE Magazine.
Just fill out the enclosed reply card and return it to us. You can also use the card to order any other information by post.
Thank you!

You can also give us your feedback online.
For further information, visit:
www.dspace.com/magazine
Release information on dSPACE products is available at:
www.dspace.com/releases

Advanced Driver Assistance Systems – Get There Safely with dSPACE

Can you fulfill rising customer expectations and ever-tougher safety requirements for advanced driver assistance systems – all without development costs spiraling out of control? Of course you can. With a well-coordinated tool chain for function development, virtual validation and hardware-in-the-loop simulation, in which perfectly matched tools interact smoothly throughout all the development steps. Whether you're integrating camera and radar sensors, modeling vehicles and traffic scenarios, or running virtual test drives.

Get your advanced driver assistance systems on the road – safely!

Embedded Success dSPACE